



**HAL**  
open science

# PerfectO: An Online Toolkit for Improving Quality, Accessibility, and Classification of Domain-based Ontologies

Amelie Gyrard, Ghislain Auguste Ateazing, Martin Serrano

## ► To cite this version:

Amelie Gyrard, Ghislain Auguste Ateazing, Martin Serrano. PerfectO: An Online Toolkit for Improving Quality, Accessibility, and Classification of Domain-based Ontologies. *Semantic IoT: Theory and Applications Interoperability, Provenance and Beyond*, 2021, 10.1007/978-3-030-64619-6\_7. hal-03200754

**HAL Id: hal-03200754**

**<https://hal.science/hal-03200754>**

Submitted on 16 Apr 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Perfect0: An Online Toolkit for Improving Quality, Accessibility, and Classification of Domain-based Ontologies

Amélie Gyrard<sup>14\*\*</sup>, Ghislain Ateazing<sup>2</sup>, and Martin Serrano<sup>3</sup>

<sup>1</sup> Kno.e.sis, Wright State University, USA, [amelie@knoesis.org](mailto:amelie@knoesis.org),

<sup>2</sup> Mondeca, 35, Boulevard Strasbourg, 75010, Paris, France

[ghislain.ateazing@mondeca.com](mailto:ghislain.ateazing@mondeca.com)

<sup>3</sup> Insight Center for Data Analytics, National University of Galway, Ireland

[martin.serrano@insight-centre.org](mailto:martin.serrano@insight-centre.org)

<sup>4</sup> Trialog, Paris, France

**Abstract.** Sensor-based applications are increasingly present in our everyday life. Due to the enormous quantity of sensor data produced, interpreting data and building interoperable sensor-based applications is needed. There are several problems to address the heterogeneity of 1) data format, 2) languages to describe sensor metadata, 3) models for structuring sensor datasets, 4) reasoning mechanisms and rule languages to interpret sensor datasets, and 5) applications. Semantic Web technologies (a.k.a, knowledge graphs), are immersed in an increasing number of online activities we perform today (e.g., search engines for gathering information). There is a need to find better ways to share data and distribute more meaningful and more accurate information. Innovative methodologies are needed to link and associate the data from different domains to improve knowledge discovery. Semantic knowledge graphs, made of datasets and ontologies, are intended to describe and organize heterogeneous data explicitly. If an ontology is widely used to structure data of a particular domain, the accessibility and the efficiency in sharing and reusing that information will increase. For this reason, we focused on the *ontology quality* used when building sensor-based applications. We designed **Perfect0**, a **Knowledge Directory Services** tool, focusing on ontology best practices, which: 1) improves knowledge quality, 2) leverages usability, accessibility, and classification of the information, 3) enhances engineering experience, and 4) promotes engineering best practices. **Perfect0** implementation is applied to the Internet of Things (IoT) domain because it covers more than 20 application domains (e.g., healthcare, smart building, smart farm) that use sensors. **Perfect0** enhances knowledge expertise quality implemented within any ontologies as demonstrated with the Linked Open Vocabularies for IoT (LOV4IoT) ontology catalog.

---

\*\* Thanks to the Linked Open Vocabularies (LOV) team for sharing their expertise regarding the usage of validation tools.

**Keywords:** Knowledge Directory, Knowledge Directory Service, Semantic Data Interoperability, Ontology Quality, Methodology, Web of Things, Internet of Things, Semantic Web of Things, Semantic Web Technologies.

## 1 Introduction

We have been witnessing a growing number of sensors embedded in smart objects (e.g., Fitbit watches). Sensor-based applications are increasingly present in our everyday life (e.g., Mother [WR1] reminds medication, Apple HealthKit [WR2] tracks fitness, nutrition, and sleep, and Foobot measures air quality). The hardware devices, protocols, and infrastructures are already deployed. However, these devices have not been engineered by the same company nor explicitly designed to be compatible with other devices. The devices produce data that is sent to the Web to build the 'Internet/Web of Things' (IoT/WoT) applications. According to Cisco's predictions [WR3], there will be more than 50 billion of devices connected to the Internet by 2020. Due to the enormous quantity of sensor data generated, interpreting data and building interoperable IoT applications is needed. There are several heterogeneity issues to address 1) data format, 2) languages to describe sensor metadata, 3) ontologies to structure sensor datasets, 4) reasoning mechanisms and rule languages to interpret sensor datasets, and 5) applications. The challenge today is finding better ways to reuse more meaningful and more accurate information, to get useful abstractions from sensor data. Innovative methodologies are required to link the data from different domains to improve knowledge discovery.

**Semantic Web technologies** are widely used in applications and in online activities that we perform every day. Google, reused the **Knowledge Graph** (KG) [WR4] term in 2012, which became popular, and demonstrated the impact of semantic web technologies. Everyday, we are using KG technologies without being aware of it. Indeed, when we are looking for information (e.g., a famous person, a restaurant) using the Google search engine, structured information appears on the right. According to Paulheim's KG survey [1], a KG (1) mainly describes real-world entities and their interrelations, organized in a graph, (2) defines possible classes and relations of entities in a schema (e.g., ontologies), (3) allows for potentially interrelating arbitrary entities with each other, and (4) covers various domains. KGs are based on Linked Data [2] mechanisms implemented by semantic web standards (i.e., XML, RDF, OWL, SKOS, SPARQL, etc.) to enable: (1) large-scale data integration, and (2) reasoning on information over the Web. Linked Data structures data according to ontologies. Ontologies [3] facilitate data exchange and interoperability within applications by modeling a specific domain. Many academic and industrial communities now understand the need for ontologies. Recently, authors at [4] define a KG as a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities. The graph of data (aka data graph) conforms to a graph-based data

model, which may be a directed edge-labelled graph, a property graph, etc.

The design, integration, and re-usability of ontologies are widely encouraged by the semantic web community. However, in other communities (e.g., IoT, Sensor Web, Semantic Sensor Networks), reusing ontologies is still a significant challenge where most of the time, similar ontologies are re-designed regularly, and not aligned with existing ontologies. There is a need to reference and classify those ontologies. Ontology catalogs such as Linked Open Vocabularies (LOV) [5], LOV for the Internet of Things (LOV4IoT) [6] [7] [8], and BioPortal for medical ontologies encourage the reuse of ontologies. However, interoperability issues remain due to the lack of proper guidance to acquire knowledge for best practice and the lack of time.

Helping ontology designers in improving ontologies is essential for the following reasons: (1) Reusing domain knowledge in other projects/domains because most of the time the knowledge is already conceived and implemented as ontologies. (2) Reducing the exhaustive job of re-inventing ontologies. (3) Reducing development time. (4) Increasing semantic interoperability among systems when machines share the same ontology. (5) Providing cross-domain interoperability, the same ontology can be used in different domains. For instance, a smart home ontology is relevant in healthcare for Ambient Assisted Living (AAL) to help elderly people to stay independent at home.

Improving the quality of knowledge representation is gaining more attention, existing surveys are: 1) knowledge graphs in [1], 2) Linked Data in [9], and 3) ontology quality in [10] [11] [12][13]. Fernandez-Lopez et al. [14] study the LOV ontology catalog and demonstrate that: (1) 36% of the ontologies registered in LOV are not appropriately be loaded, (2) 78% of ontologies reuse W3C endorsed ontologies, and (3) 27% of the LOV ontologies do not have English-language identifiers such as labels and comments, or within URIs. Easy-to-use ontology improvement tools are still needed to ease ontology designers' life. We focus on the ontology quality by foster a subset of tools to encourage ontology reuse.

**Contributions:** Our proposed solution, called **Perfect0** (stands for Perfect Ontology), relies on organizing and helps improve the existing knowledge rather than generating more. **Perfect0**'s primary objective is to increase the ontology engineer/designer experience and facilitate the ontology quality improvement process by selecting a subset of useful validation tools providing web services or online GUIs. **Perfect0** enhances ontology experience, accessibility, ontology classification, ontology interlining, and the optimization of the available ontologies. In this chapter, we use **Perfect0** implementation in the Internet of Things domain enhancing ontologies from Linked Open Vocabularies for IoT (LOV4IoT) ontology catalog as a demonstration of how **Perfect0** can be used beyond the semantic web domain. **Perfect0** LOV4IoT use case is included, and it is de-

ployed as the proof of concept and experimental validation in other fields.

**Structure of the chapter:** The rest of the chapter is structured as follows: Section 2 highlights the need to help ontology designers in improving ontology quality, accessibility, and knowledge classification. Section 3 investigates the literature to enhance ontologies by covering complementary research topics: quality, evaluation, methodologies, and metrics. Section 4 describes **Perfect0**, a web-based tool to improve ontology quality and its implementation. Section 5 provides evaluation, lessons learned, and discussion. Section 6 concludes the chapter and highlights future work.

## 2 Improving Ontology Quality, Accessibility and Knowledge Classification

Following the above explanations, we introduce the **Perfect0** methodology, which promotes ontology best practices to leverage quality and usability, semantic interoperability, and classification. **Perfect0** can be defined by the term of *Knowledge Directory Services* [15] that we enrich with the definition as follows: (1) encouraging the reuse of knowledge through ontology catalogs, (2) disseminating best practices to improve ontology quality, (3) referencing ontology validation tools to enhance ontology usability, and (4) facilitating semantic interoperability among ontologies.

### 2.1 Encouraging the reuse of knowledge through ontology best practices

The reuse of ontologies has always been the primary problem in ontology engineering. Tools and repositories where ontologies can be indexed are limited. Existing ontology catalogs need to be widely disseminated. We take inspiration from software engineering and programming languages that define best practices when developing such as starting a class with an uppercase, providing comments, documentation, etc. Technical aspects are essential (e.g., the Java programming language community provides certification to encourage programmers to follow best practices). There are also code repositories such as GitHub and BitBucket where developer community members are invited to reuse the code.

Similarly, the design of **Perfect0** takes inspiration from the software engineering world to disseminate best practices for *Semantic Interoperability* [16] where there are already ontology engineering research fields working on quality such as *ontology quality*, *ontology methodology*, *ontology evaluation*, etc. **Perfect0** enhances self-assessment for ontology improvement from technology, and also human perspective.

*Improving the quality of the LOV4IoT ontology catalog* LOV4IoT [WR5] [8] is an ontology catalog referencing 499 ontology-based IoT projects (in March

2019) integrating semantic web technologies. LOV4IoT covers more than 20 application domains: robotics, smart cities, Internet of Things (IoT), Web of Things (WoT), Ambient Assisted Living (AAL), wearable computing and Wireless Body Area Networks (WBANs), agriculture, smart energy, water management, waste management, environment, logistics, manufacturing, weather, home, tourism, healthcare, etc. The healthcare domain is the main exception being familiar with semantic web best practices, but it is not the case for other communities such as the smart home. LOV4IoT aims to encourage the reuse of existing IoT ontologies. LOV4IoT extends Linked Open Vocabularies (LOV) [5] because multiple ontologies that have been designed do not meet the best practices criteria preconized by the LOV community.

The experience acquired when referencing new ontologies on the LOV4IoT ontology catalog has been used to create the **Perfect0** ontology improvement methodology, explained in Section 4. Only a subset of 221 ontologies is available online that are integrated with **Perfect0**. Most of the ontologies after improvements will become more interoperable to be referenced in the LOV catalog. LOV4IoT is an ontology catalog incubator for those ontologies developed by multi-domain communities.

## 2.2 Disseminating tools for ontology quality

There is a real need for the proper tools to improve ontology quality, guide, and educate ontology designers/engineers. A state of the art analysis (Section 3) for best practices, quality, and evaluation of ontologies has been done to fill this gap. Taking inspiration from the Linked Data blog [WR26], we innovatively disseminate state of the art: a website which encourages semantic interoperability by classifying and referencing reusable tools that can assist the improvement of ontologies.

*Example for dissemination and ontology quality* Federated Interoperable Semantic IoT Testbeds and Applications (FIESTA-IoT)<sup>5</sup> is an H2020 EU project focusing on IoT semantic interoperability. To integrate new testbeds (e.g., smart cities), a set of practices needs to be followed to semantically annotate and link the data, deduce new knowledge from data, and unify models, services, and applications. FIESTA-IoT ontology [17], aims to integrate a set of IoT-related ontologies that have been studied: 19 ontologies for IoT, 28 ontologies for Wireless Sensor Networks (WSNs), 8 ontologies for the Web of Things (WoT), and 16 ontologies for smart cities. Our goal was to design a *unified IoT ontology* by reusing well-designed and most common ontologies. Furthermore, the FIESTA-IoT dataset validation tool compliant with the FIESTA-IoT ontology has been implemented [18] to check interoperability among semantic datasets. When IoT-related ontologies have been found (e.g., reading scientific publications), we have suggested the ontology URLs to the LOV ontology catalog. We have learned from the LOV community (through email exchanges) that the required practices are not followed most of the time and ontologies need to be improved with the set

<sup>5</sup> <http://fiesta-iot.eu/>

of tools presented in Table 3. From this experience, we realize the need to disseminate those practices and tools within the IoT community [19]. In the LOV journal publication [5], note the special acknowledgment for our dissemination work<sup>6</sup>. Further, there is a common co-author. Evaluating ontology quality helps decide which ontologies to reuse.

### 2.3 Development Time Optimization and Ontology Improvement

Re-using ontologies is highly encouraged by the ontology engineering research community [20]. What the communities urgently need, is an *ontology improvement methodology* that assists novice ontology designers in improving and publishing reusable ontologies. The key benefit is to enhance ontology methodology development [21], with a focus of best practices and help in the learning curve to reduce the development time. It can be done by integrating existing tools. The main novelty is to extend our analysis [19] on improving ontologies and automate the process of the usage of the tools.

### 2.4 Semantic Ontology Interoperability Methodology

IoT infrastructure requires information sharing for identifying, locating, organizing, and managing everyday "things", their services, and resources. The IoT data infrastructure relies on Big Data systems to have enough capacity to store and process the amount of information collected [22]. SEG 3.0 is a methodology defined to enable data interoperability across different data systems, software platforms, and applications [23]. SEG 3.0 implements data exchange using an ontological approach and allowing access to data. **Perfect0** is a platform designed to reduce the learning curve to support the SEG 3.0 semantic interoperability. **Perfect0** (explained in Section 4) focuses on ontological improvement and interoperability.

*Semantic interoperability according to standards* **IERC AC4 (European Research Cluster on the Internet of Things)** [24] [25] [16] highlighted four levels of interoperability: (1) technical, (2) syntactic, (3) semantic, and (4) organizational. Technical and syntactical interoperability were the main concerns in research and development in recent years. IERC AC4 does not reference concrete tools encouraging - (i) the best practices, (ii) the use of methodologies to ensure interoperability among ontology-based IoT applications, and (iii) reuse of the domain knowledge already designed within ontologies. For this reason, **Perfect0** provides a set of concrete tools to encourage semantic interoperability and reuse improved ontologies. Standardizations are demonstrating the need to help ontology developers, and ontology quality to achieve semantic interoperability. **AIOTI Working Group 3** is dedicated to IoT standardization and has confirmed that one of the most important topics are the semantic interoperability<sup>7</sup>. **OneM2M, an international standard for IoT and Machine to**

<sup>6</sup> "Thanks to Amelie Gyrard for the help on the project"

<sup>7</sup> <https://ec.europa.eu/digital-single-market/en/alliance-internet-things-innovation-aioti>

**Machine (M2M)**, is looking for the best IoT semantic interoperability practices as well [26]. **Semantic Interoperability for the Web of Things White paper** highlights the main interoperability issues [27] and citing our research work as a baseline. Furthermore, in October 2019, two "Semantic Interoperability for IoT" White Papers [28] [29] have been released and disseminated by ETSI, W3C, AOITI, etc. to guide IoT and standard developers (also from OneM2M and ISO) to reuse and develop semantics-based IoT applications easily.

### 3 Related Work

We review the literature in this section which is summarized in Table 1. The literature review has been introduced earlier when the *Catalogs of Tools* depicted in Figure 9 has been described in Section 8.1. Existing surveys are classified in Section 3.1, ontology methodology in Section 3.2, ontology evaluation in Section 3.3, ontology metrics in Section 3.4, and ontology quality in Section 3.5. Finally, the main limitations of the related work are highlighted in Section 3.6.

#### 3.1 Existing Surveys

Existing surveys are covering complementary research topics for ontology quality, evaluation, ranking, metrics, usability, and methodologies that are reviewed within this section and summarized within Table 1.

McDaniel et al. [10] provide a set of metrics to evaluate ontologies, but it can be challenging to implement them. For instance, the recognition metric computes the number of times the ontology is downloaded which is not provided by catalogs such as LOV and LOV4IoT. A second example is the lack of explanations for the metric consistency implementation. Reasoners can explain if the consistency metric is satisfied but do not provide an explicit number. McDaniel et al. design the **Domain Ontology Ranking System (DoORS)** prototype [10] to query ontology catalogs with specific keywords and assess ontology quality. Metrics are implemented to automate the selection of the ontology within the prototype. The DoORS prototype provides the following quality assessment metric modules: (1) *Syntactic*: Quality, lawfulness, richness, structure, (2) *Semantic*: quality, consistency, interpretability, precision, (3) *Pragmatic*: quality, accuracy, adaptability, comprehensiveness, ease of use, relevance, (4) *Social*: quality, authority, history, recognition, and (5) *Overall*. We are expecting more explanations for why and how those metrics have been chosen, and the meaning of the evaluation numbers provided within the prototype.

However, this excellent survey paper, published in 2018, misses important references of pioneer work to define metrics (e.g., the authors in [30] outline metrics such as the ontology competency and completeness). Furthermore, compared to their work, we collect a set of tools to improve ontologies. We took into consideration others metrics such as documentation, visualization, dissemination on ontology catalogs. Since those tools have most of the time ontology validators integrated, it enables to evaluate ontology quality at the same time.

**Ma et al.** [31] propose the **Ontology Usability Scale (OUS)**, a ten-item Likert scale derived from statements prepared according to a semiotic framework and an online poll in the semantic web community to evaluate ontology usability. ISO 9241-11 defines usability as follows: "the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency, and satisfaction in a specific context of use." The authors estimate the costs of using ontologies when developing applications to create better ontologies. An ontology may be consistent (i.e., without any contradictory assertion, complete (i.e., without any missing definition) and concise (i.e., without any unnecessary definition) but still unusable or very cumbersome to use (e.g., due to lousy documentation). The authors classify ontologies into three categories: (1) pragmatics, (2) semantics, and (3) syntax.

**Raad et al.** [11] investigate what makes a good ontology by analyzing ontology evaluation methods and discuss their advantages. These methods are used to evaluate the quality of automatically constructed ontologies. A good ontology can contribute to the success of semantic services and various knowledge management applications. Four categories have been designed: (1) gold standard-based, (2) corpus-based, (3) task-based, and (4) criteria-based.

**Hlomani et al.** design competency questions for ontology evaluation with a focus on quality and correctness as follows [12]: (1) Does the model cover required context information? (2) Does the DL language provide the logical constructs required by the reasoner? (3) Are the adaptation purposes sufficient to describe a WoT app context?, and (4) Are the adaptation purposes redundant or overlapping?

**Corcho et al.** survey the existing ontology methodologies [32] and answer the following questions: (1) which methods and methodologies can I use for building ontologies?, (2) which tools give support to the ontology development process, and (3) which languages can I use to implement the ontology? This work can be used as a guideline for analyzing the existing methodologies and tools for ontology engineering but need to be updated since it has been published in 2003.

### 3.2 Ontology Methodologies and Ontology Design Patterns (ODPs)

Ontology methodologies encourage well-designed ontologies and their reuse. Reusing ontologies is challenging. Sometimes several ontologies are using the same concept that the ontology designers need. More guidance needs to be provided to assist in the selection of ontologies. We reviewed existing *ontology methodologies* (also referenced in Table 1). The NeON [20] and Noy et al.'s [21] methodologies are the most popular ones.

The Noy et al.'s **ontology development 101 methodology** encourages ontology designers to reuse existing domain knowledge (e.g., ontologies) [21]. The methodology consists of the following iterative steps: (1) determination of the domain and scope of the ontology, (2) reuse of existing ontologies, (3) enumeration of essential terms, (4) definition of the classes and the class hierarchy, and (5) definition of the properties and creation of instances.

The **Neon** project<sup>8</sup> recommends reusing available knowledge and proposes a set of methodologies [33]. The Neon project focuses on nine scenarios [33]: (1) from specification to implementation, (2) reusing and re-engineering non-ontological resources, (3) reusing ontological resources, (4) reusing and re-engineering ontological resources, (5) reusing and merging ontological resources: ontology matching tools enable ontology aligning or merging, (6) reusing merging, and re-engineering ontological resources, (7) reusing ontology design pattern (ODPs), (8) restructuring ontological resources, and (9) localizing ontological resources to translate of all the ontology terms into another natural language. We are mainly interested in scenario 3 to help IoT developers in reusing ontologies relevant for IoT. The other future steps are interesting for re-designing ontologies in an interoperable manner and not "reinventing the wheel at each ontology development" to speed up the ontology development process.

**On-to-Knowledge** is another methodology for designing ontologies comprised of four steps: (1) kick-off, (2) refinement, (3) evaluation, and (4) ontology maintenance [34].

*Ontology methodologies* such as NeOn [33] and METHONTOLOGY [35] have been taken into consideration when designing PerfectO since the methodologies highly encourage to document and reuse ontologies. For instance, METHONTOLOGY does not provide a set of tools. PerfectO enriches those methodologies with a set of tools (presented in Section 8.3) to improve ontologies. Our approach is to reuse as much as possible the existing tools by integrating them instead of redesigning a new tool from scratch.

*Ontology Design Patterns* is a research topic itself, an entire book is dedicated to this topic [36].

### 3.3 Ontology Evaluation

There is a need for tools supporting *ontology evaluation* research approaches, also referenced in Table 1.

**Gruninger et al.** [30] are the pioneer working on ontology evaluation in 1995 and define the competency of the ontology as a set of questions that an ontology answers. The competency of the ontology can be evaluated by proving completeness theorems for the competency questions.

**Vrandečić et al.** [46] design a conceptual framework for ontology evaluation to assess the quality of an ontology for the Web. Eight criteria have been defined which are accuracy, adaptability, clarity, completeness/competency, computational efficiency, conciseness, consistency/coherence and organizational fitness/commercial accessibility. Evaluating ontologies can be done to deal with six different aspects: vocabulary, syntax, structure, semantics, representation, and context. The implementation of the work has been done within the Semantic MediaWiki, an extension of the MediaWiki which provides collaborative creation and maintenance of ontologies. Another innovative idea is the introduction of unit tests for ontologies by the same authors [47].

<sup>8</sup> <http://www.neon-project.org/>

Authors, Framework & Publication	Year of Publication	Topic	Prototype URL
McDaniel et al. [10] Domain Ontology Ranking System (DoORS)	2018	Ontology <b>Quality</b> Survey Ontology <b>Metrics</b> Survey Ontology <b>Ranking</b> Survey	✓URL [WR29]
Ma et al. [31] Ontology Usability Scale (OUS)	2018	Ontology <b>Usability</b> Survey	✗
Paulheim et al. [1]	2017	<b>Knowledge Graph Evaluation</b> Survey	✗
Hitzler et al. [36]	2016	<b>Ontology Design Patterns</b>	✓ODPs wiki [WR30]
Zaveri et al. [9]	2015	Linked <b>Data Quality Survey</b>	✗
Raad et al. [11]	2015	Ontology <b>Evaluation</b> Survey	✗
Poveda et al. [37]	2014-2010	Ontology <b>Quality</b>	✓URL [WR8a] Integrated with PerfectO
Hlomani et al. [12]	2014	Ontology <b>Evaluation</b> Survey	✗
Staab et al. [34] On-to-Knowledge	2013	Ontology <b>Methodology</b>	✗
Duque-Ramos et al. [38] [39] OQuaRE	2013	Ontology <b>Quality</b>	✓URL [WR25]
Suarez et al. (NeON) [33]	2012	Ontology <b>Methodology</b>	✓URL [WR24]
Garcia et al. [13]	2010	Ontology <b>Metrics</b> Survey	✗
Fernandez et al. [40] METHONTOLOGY [35]	2009 1997	Ontology <b>Quality</b> Ontology <b>Methodology</b>	✗
Tartir et al. [41] [42] OntoQA	2007-2005	Ontology <b>Quality</b> 3 Ontology <b>Metrics</b> 9 Instance <b>Metrics</b>	✗
Brank et al. [43]	2005	Ontology <b>Evaluation</b> Survey	✗
Burton et al. [44]	2005	10 ontology <b>Metrics</b>	✗
Lozano et al. [45] OntoMetric	2004	160 ontology <b>Metrics</b>	✗
Corcho et al. [32]	2003	Ontology <b>Methodology</b> Survey	✗(Survey paper)
Noy et al. [21] Ontology development 101	2001	Ontology <b>Methodology</b>	✗
Gruninger et al. [30]	1995	Ontology <b>Evaluation</b>	✗

Table 1. Summary of ontology methodology, evaluation, metric, and quality works

### 3.4 Ontology Metrics

*Ontology metrics* have been defined in [10] [12]. However, there is a need to classify and prioritize metrics. Ontology metric works are also referenced in Table 1.

**OntoMetric** is a methodology to choose the appropriate ontology for a specific system [45]. Five dimensions are provided: (1) the content of the ontology and its organization, (2) the language used for the implementation, (3) the development methodology employed to build the ontology, (4) the software tools used to design the ontology, and (5) the costs to develop and maintain the ontology. The OntoMetric framework defines 160 characteristics according to those five dimensions.

**Burton et al.** [44] define ten metrics for ontology quality: lawfulness, richness, interpretability, consistency, clarity, comprehensiveness, accuracy, relevance, authority, and history.

### 3.5 Ontology Quality

Tools are emerging to evaluate the *ontology quality* (also referenced in Table 1), but not enough known outside from the semantic web community. There is a real necessity to disseminate those tools and test their usability.

**Fernandez et al.** [40] define the term of "good" and high-quality ontology which means the easy reuse of some parts of a given ontology instead of the entire ontology. "Parts of ontologies" refer to Ontology Design Patterns (ODPs) [48] and modular ontologies [49] research approaches (not covered in this paper).

**OQuaRE** is a framework for evaluating the quality of ontologies [39] based on the SQuaRE standard for software quality evaluation. A quality model and quality metrics (structural, functional adequacy, reliability, operability and maintainability) have been defined. The framework has been evaluated with units of measurement ontologies. Future work of this paper highlights the needs of automated ontology evaluation.

**OntoQA** [41] [42] assists ontology developers and users to determine the quality of an ontology. It provides metrics to evaluate ontology design and instances. OntoQA provides three ontology metrics: (1) *Relationship richness*: an ontology that contains many relations other than class-subclass relations is more precious than a taxonomy with only class-subclass relationships, (2) *Attribute richness*: the number of attributes that are defined for each class can indicate both the quality of ontology design and the amount of information of instance data, and (3) *Inheritance richness*: a good indication of how well knowledge is grouped into different categories and subcategories in the ontology. OntoQA defines nine instance metrics: (1) *Class richness* for KB Metrics is related to how instances are distributed across classes, (2) *Average population* for KB Metrics (average distribution of instances across all classes) indicates the number of instances compared to the number of classes. It can be useful if the ontology developer is not sure if enough instances were extracted compared to the number of classes, (3) *Cohesion* for KB Metrics can be used to indicate what areas need more instances to connect instances more closely, (4) *Importance* for class metrics provides the percentage of instances that belong to classes at the subtree rooted at the current class for the total number of instances, (5) *Fullness* for class metrics is mainly used by an ontology developer interested in knowing how well the data extraction was with respect to the expected number of instances of each class, (6) *Inheritance richness* for class metrics indicates how well knowledge is grouped into different categories and subcategories under this class, (7) *Relationship richness* class metrics: measures how much of the properties in each class in the schema is being used at the instance level, (8) *Connectivity* for class metrics explains which classes play a more central role than other classes, and (9) *Readability* for class metrics indicates the existence of human-readable de-

scriptions in the ontology, such as comments, labels or captions. From our point of view, is also really relevant for doing automation with user interfaces.

OntoQA provides three kinds of evaluations: (1) evaluation-based validation, (2) symbolic-based validation, and (3) attribute-based validation. OntoQA has been evaluated with three ontologies: SWETO, TAP and GlycO, and related datasets. As a future work, the need for a web-based tool to automatically measure the quality of the ontologies is explained. However, we did not find such tools available online that we can reuse and integrate with other tools.

### 3.6 Limitations of the Related Work

PerfectO and **Widoco** [50] (a tool not introduced yet) have the common objective of helping developers to create more reusable ontologies. Widoco is an open-source ontology documentation standalone Java application. Widoco is not integrated yet within our PerfectO approach since it does not provide a web service to ease a fast development. Widoco focuses on ontology documentation which refers to one step in our ontology improvement methodology explained in Section 4.2.

The current literature focused on methodologies or validation tools (used in Table 3) to design good ontologies. We did not find any work surveying the existing web-based tools and designing the methodology which integrates current web services to support better the design of reusable ontologies. Our PerfectO approach aims to assist the LOV project by providing a web site to provide more guidance to help ontology designers fixing ontology errors encountered with the validation tools.

## 4 PerfectO: Architecture and Implementation

The PerfectO architecture and its components are described in this section. PerfectO demonstrates its applicability and validated experience for enhancing ontologies on a particular use case for the Internet of Things.

### 4.1 PerfectO Architecture

The PerfectO architecture (depicted in Figure 1) follows interoperability principles established with the intention to exchange data with common ontologies [23]. The architecture comprises GUIs, APIs, and core components. Ontology designers interact with our web-based software using Graphical User Interfaces (GUIs). Application Programming Interfaces (APIs) are either employed by the GUI or by the ontology designers and are developed according to RESTful principles [51]. The core components of the architecture are as follows:

1. **Ontology improvement methodology** encourages ontology designers to improve the ontology (explained in Section 4.2). Improvement methodologies for datasets, queries or reasoning mechanisms are left for future work.

2. **Catalogs of Tools** refers to state of the art (see Section 3). State of the art is also shared online through clickable links and interactive mind maps<sup>9</sup> (introduced in Section 8.1).
3. **Dr. PerfectO Availability of Validation Tools (DPAT)** checks the availability of tools (see Section 8.3).
4. **PerfectO Guidance** selects a subset of validation tools (introduced in Section 8.3).

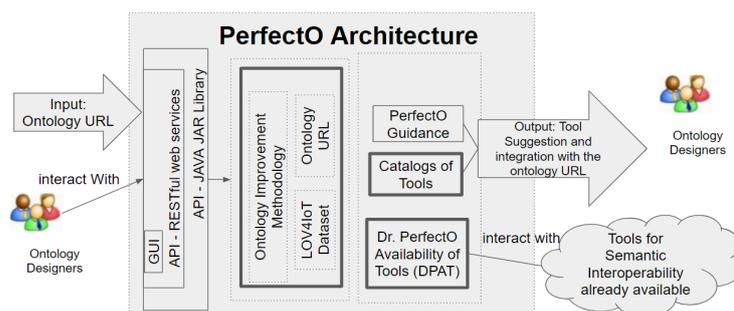


Fig. 1. PerfectO Architecture

**Technologies used for the implementation:** The RESTful web services have been implemented in Java using the Jersey web service library. For instance, the web service<sup>10</sup> returns the result status OK or NOT OK. The URL parameter is a web service pinging the tools mentioned in Section 8.3. The web services are queried using AJAX technology. The results returned are parsed in JavaScript and displayed in the HTML web pages. We highly encourage the readers to browse the PerfectO web site<sup>11</sup> to check out the different modules already implemented.

## 4.2 Ontology Improvement Methodology

To reduce the learning curve and time-consuming development task of designing reusable ontologies, we conceived the *ontology improvement methodology* (as depicted in Figure 2). This methodology evaluates IoT and smart city ontologies (see section Evaluation [6]). The methodology comprises the following steps; some of them are interchangeable in their position:

1. **Syntactic validation** is necessary during the compilation, and the execution of ontologies with libraries to be processed by the *ontology quality*

<sup>9</sup> [http://perfectsemanticweb.appspot.com/?p=ontology\\_sota](http://perfectsemanticweb.appspot.com/?p=ontology_sota)

<sup>10</sup> <http://perfectsemanticweb.appspot.com/perfecto/statusTool/?url={url}>

<sup>11</sup> <http://purl.org/perfecto>

*methodology.* Tools such as OWL Manchester and Triple Checker can be used.

2. **Serialization** supports the OWL ontology format since it is a W3C recommendation.
3. **Interlinking** enhances interoperability, integration, and browsing among ontologies. Ontology matching tools such as LogMap<sup>12</sup> can be employed.
4. **Documentation** eases ontology understandability. Parrot and LODÉ have been chosen since web services for automatic documentation are provided. More and more tools are being designed to provide such criteria to ease the task of developers (e.g., Widoco [50]). We designed the ontology documentation mindmap<sup>13</sup> to give an overview of existing ontology documentation tools.
5. **Availability** advocates sharing the resource on the web. Developers do not have time, resources, or administrative skills to manage the server. Ideally, an ontology catalog server hosts any ontologies and provide the right URL. Sharing the ontology code and documentation on the web encourage ontology reuse.
6. **Discoverability** improves the dissemination of ontologies. For instance, suggesting ontologies on ontology catalogs and semantic search engines support the dissemination. Several ontology catalogs are available which depends on the application domain. Ideally, each ontology provides dereferenceable URI and can be tested with the Vapour tool or the Curl command line.
7. **Visualization** eases the learning phase by providing a fast understanding of the ontology. WebVOWL tool is integrated to provide an automatic ontology graph visualization.
8. **Ontology Consistency** can be improved with tools like Oops! tool or SHACL<sup>14</sup> validation rules which detects numerous ontology pitfalls and suggests how to avoid/fix them.

The PerfectO's Ontology Improvement Check-Up list is not intended to be used in the order of the steps/operations specified in the Fig.2. Two typical workflows in real-word settings are the following:

- Steps order: 1, 2, 8, 3, 4, 5, 6 and 7
- Steps order: 1, 2, 8, 4, 3, 5, 6 and 7

Steps 3 or 4 can be easily done at any order after syntactic validation, serialization and consistency checking. Note here step 8 at the early process helps detecting errors also very earlier, specially for newbies in ontology engineering.

*PerfectO Ontology Improvement Tool Integrated with LOV4IoT.* We have implemented the *ontology improvement* component<sup>15</sup> by integrating six tools

<sup>12</sup> <https://www.cs.ox.ac.uk/isg/projects/LogMap/>

<sup>13</sup> [http://perfectsemanticweb.appspot.com/?p=ontology\\_sota#div\\_ontology\\_documentation\\_mindmap](http://perfectsemanticweb.appspot.com/?p=ontology_sota#div_ontology_documentation_mindmap)

<sup>14</sup> <https://www.w3.org/TR/shacl/>

<sup>15</sup> <http://perfectsemanticweb.appspot.com/?p=ontologyValidationLOV4IoT>



**Fig. 2.** PerfectO's Ontology Improvement Check-Up Levels

(TripleChecker, Oops!, Parrot, Vapour, LOD, and WebVOWL) with the ontologies referenced within the LOV4IoT RDF dataset<sup>16</sup>.

Figure 3 demonstrates the implementation of the PerfectO *Ontology Improvement* methodology (explained within the Architecture Section 4.1) integrated with the LOV4IoT ontology catalog. Figure 3 comprises three main parts:

1. A drop-down list is shown with all IoT applicative domains referenced within the LOV4IoT ontology catalog (more than 20 domains) introduced in Section 2.1. Once a domain is selected, the second drop-down list is filled in. In this example, the selected domain is IoT.
2. A second drop-down list provides all ontologies referenced within LOV4IoT for a specific domain. In this example, all ontologies for IoT have a URL which is shown within the tooltip.
3. Once a particular ontology is selected, tools mentioned above in section 8.3 are automatically integrated. The integrated tools appear on the right part of the screen. Clickable links are provided to go to the web service tools with the selected ontology.

*PerfectO Ontology Improvement Tool with an Ontology URL.* A more sophisticated interface for ontology validation has been designed where the designers

<sup>16</sup> <http://purl.org/lov4iot-dataset>

Choose a domain (e.g., Internet of Things (IoT)):

Internet of Things (IoT) **1**

Once you select a domain, all ontologies for this specific domain will be queried: Then, select one ontology in the second drop-down list (e.g., FIESTA-IoT) On the right will appear clickable links for the integration with validation tools. **2**

IoT-A ontology - Temperature Sensor Service - IoT-A project - Service Modeling for the Internet of Things [De et al. 2011]  
 IoT-A ontology - OWL IoT Service - IoT-A project - Service Modeling for the Internet of Things [De et al. 2011]  
**FIESTA-IoT ontology - FIESTA-IoT project - Unified IoT Ontology to Enable Interoperability and Federation of IoT ontology - Ontologies for the Internet of Things [Hachem et al. 2011]**  
 MOFI ontology - OMN Monitoring Unit - Fed4FIRE Project - Towards Semantic Monitoring Data Collection an  
 MOFI ontology - OMN Monitoring Data - Fed4FIRE Project - Towards Semantic Monitoring Data Collection an  
 IoT-Test and Sensei project and ontology - A comprehensive ontology for knowledge representation in the inte  
 FIIPA-Device Ontology - FIIPA-Device Ontology  
 IoT-O Ontology - Semantique et Internet des Objets: d un etat de l art a une ontologie modulaire [Seydoux et  
 Smart City PRISMA ontology - A Smart City Data Model based on Semantics Best Practice and Principles [C  
 WoT ontology - Web of Things Ontology according to the W3C Interest Group. [Poveda Villalon et al. (Onto  
 OneM2M ontology - OneM2M standard  
 Mobile ontology - Towards a Standardized semantic model for the mobile domain [Villalonga et al. 2009].  
 IoT-A ontology - Entity Model - IoT-A project - Service Modeling for the Internet of Things [De et al. 2011]  
 IoT-A ontology - Resource Model - IoT-A project - Service Modeling for the Internet of Things [De et al. 2011]

Loading the ontology selected with WebOWL for automatic graph visualization **3**  
 Loading the ontology selected with Parrot for automatic documentation (Choice 2)  
 Loading the ontology selected with Triple Checker for syntax checking and much more  
 Loading the ontology selected with OOPS to improve the design of the ontology  
 Loading the ontology selected with Vapour to check deferring URI and content negotiation  
 Loading the ontology selected with LODÉ for automatic documentation (Choice 1)

Be careful, sometimes some ontologies cannot be loaded with the tools!  
 This is the purpose of PerfectO, understand why and see how the ontologies can be fixed.

**Fig. 3.** The PerfectO ontology improvement methodology implemented and integrated with the LOV4IoT ontology catalog

**Enter the URL of your ontology:**

E.g., <http://purl.org/iot/ontology/fiesta-iot#> or <https://w3id.org/seas/> or <https://www.irit.fr/recherches/MELODI/ontologies/IoT-O>

Click on the button, to check your ontology with validation and improvement tools below:

[Click here to improve your ontology with all of those tools: WebVOWL, Parrot, TripleChecker, Oops, LODÉ, Vapour, etc.](#)

Sometimes some ontologies cannot be loaded with those tools! This is the purpose of PerfectO, understand why and see how the ontologies can be fixed. For your information, the previous version of this tool is still available here.

Name Tool and Logo	Tool Classification	Tool Description	Tool availability status	Integration with this tool	Status on PerfectO comments
	Visualization	WebOWL provides an automatic graph visualization		Your ontology integrated with WebVOWL here!	Tool already here in V1
	Syntactic Validation	Triple Checker provides syntax checking and much more.		Your ontology integrated with TripleChecker here!	Tool already here in V1
	Documentation	LODE provides an automatic documentation (Choice 1).		Your ontology integrated with LODÉ here!	Tool already here in V1
	Discoverability	Vapour checks deferring URI and content negotiation.		Your ontology integrated with VAPOUR here!	Tool already here in V1

**Fig. 4.** The PerfectO ontology improvement methodology implemented using an ontology URL - Part I

	Ontology Consistency	OOPS improves the design of the ontology.		Your ontology integrated with OOPS here!	Tool already here in V1
	Documentation	Parrot provides automatic documentation (Choice 2).		Your ontology integrated with Parrot here!	Parrot hosted on Mondeca server but takes time. Be patient!
	Discoverability	Linked Open Vocabularies (LOV) is an ontology catalog. Suggesting your ontology on ontology catalogs will improve its dissemination.		Your ontology integrated with LOV SUGGEST here!	Tool added in V2 - April 2018
	Discoverability	BioPortal is an ontology catalog for healthcare. Suggesting your ontology on ontology catalogs will improve its dissemination.		Your ontology integrated with BIOPORTAL here!	Tool added in V2 - April 2018 Need login and password for automatic integration
	Discoverability	Linked Open Vocabularies for Internet of Things (LOV4IoT) is an ontology catalog for Internet of Things (IoT) which includes more than 20 applicative domains (smart cities, smart home, robotics, environment, etc.) Suggesting your ontology on ontology catalogs will improve its dissemination.		Your ontology integrated with LOV4IoT here!	Tool added in V2 - April 2018
	Interlinking	LogMap is an ontology matching tool.		Your ontology integrated with LOGMAP here!	Tool added in V2 - April 2018 Need name, institution and email for automatic integration

Fig. 5. The PerfectO ontology improvement methodology implemented using an ontology URL - Part II

directly enter the ontology URL in case the ontology is not referenced yet within the catalog. Figure 4 and show an extract of the interface which is online<sup>17</sup>:

1. The user enters a specific ontology URL or can use the one by default.
2. The user clicks on the button to automatically integrate the ontology with the tools mentioned in section 8.3. A table is created to integrate the tools with the given ontology automatically. The tools are accessible through the clickable links thanks to their web service. The generated table enables quick access to any tools to improve the ontology.

Figure 4 and show a table which provides the name of the tool in the first column. The tool classification is provided in the second column, and the classification is introduced in our proposed PerfectO methodology explained in Section 4. The third column describes the usage of the tool. The fourth column checks the availability of the web service. The fifth column integrates the ontology to improve with the specific tool. The last column provides some comments regarding the integration of the tools (e.g., some maintenance issues are known, new tools recently added).

### 4.3 Use Cases

Semantic Interoperability for the Web of Things White paper highlights the main interoperability issues [27] and citing our research work as a baseline. Furthermore, in October 2019, two "Semantic Interoperability for IoT" White Papers [28] [29] have been released and disseminated by ETSI, W3C, AOITI, etc. to guide IoT and standard developers (also from OneM2M and ISO) to reuse and develop semantics-based IoT applications easily. The FIESTA-IoT ontology [17] has been employed within the FIESTA-IoT H2020 EU project involving 14 academic and industrial partners, it enables to process data generated by sensor within smart cities. The LOV4IoT dataset is used within experiments to rank the popularity of IoT ontologies in [52] and consider the following metrics: availability, believability, understandability, interlinking, PageRank, consistency, and richness. PerfectO is also refined and applied within the IEEE Autonomous robotics ontology community [53] and for well-being and affective science with the design of a recommender system for happiness which encourages the design and reuse of ontologies for emotion, food, obesity, fitness, sleep, stress, depression, acupuncture, etc. [54].

### 4.4 Limitations

The current demonstrators have some limitations. The current Graphical User Interfaces (GUI) can be more user-friendly. The demonstrators can be extended with additional tools such as the use of ontology matching tools. The overall PerfectO demonstrator is hosted on the Google Platform to avoid maintenance

<sup>17</sup> <http://perfectsemanticweb.appspot.com/?p=ontologyValidation>

and security server issues. However, latency issues can appear to load the website and query web services. Sometimes, we notice that some tools are not maintained anymore. However, we regularly update our demonstrators. For instance, Parrot has been duplicated and hosted on another server by Mondeca Labs.

Ideally, in the future, PerfectO distinguish four types of resources:

- An *Ontology* is a set of concepts and relationships between concepts to describe a specific domain. We designed the *Ontology Improvement* methodology explained in section 4.2 which is the main focus of this paper.
- A *Dataset* is structured according to ontologies.
- A *Reasoning* or inference engine executes a set of logical rules (e.g., IF THEN ELSE deductive rules) compliant with the ontologies to extract meaningful information from a dataset.
- A *Query* can be a SPARQL query compliant with the ontology and the dataset, used to retrieve a subset of the dataset required to design a specific application.

## 5 Evaluation, Lessons Learned & Discussions

The PerfectO approach is evaluated with a set of ontologies referenced by the LOV4IoT catalog in Section 5.1. Evaluation criteria defined by the Semantic Web community are reminded in Section 5.2. Discussions including lessons learned are provided in Section 5.3.

### 5.1 Ontology Quality Evaluation

We complete a detailed evaluation with **26 IoT or smart cities ontologies** referenced within the LOV4IoT ontology catalog (summarized in Table 4 and accessible online<sup>18</sup>). However, the PerfectO approach is generic enough to be applied to any ontologies (e.g., from the LOV ontology catalog). Those ontologies are tested with **six ontology improvement tools** (Parrot, WebVOWL, Oops, TripleChecker, LODÉ, and Vapour) mentioned in Table 3. Numerous ontologies cannot be successfully loaded with all of the tools and show that multiple errors are encountered. The LODÉ tool is preferred because more ontologies can be automatically documented, as compared to the Parrot tool. Furthermore, in March 2019, the Parrot automatic documentation web service does not seem maintained anymore.

As an example, a set of errors encountered with those tools:

- **[Parrot]** Unable to read input document: invalid mimeType "application/octet-stream" (returned by URI) for parrot.
- **[Parrot]** No error encountered when loaded the ontology, but nothing is displayed.
- **[Parrot]** I/O Error: Server returned HTTP response code: 403 for URL.

<sup>18</sup> [http://perfectsemanticweb.appspot.com/?p=evaluation\\_lov4iot\\_perfecto](http://perfectsemanticweb.appspot.com/?p=evaluation_lov4iot_perfecto)

- **[Parrot]** Unable to read input document: applicationrdfxml parse error: Content is not allowed in prolog.
- **[TripleChecker]** Misuse of terms from Dublin Core namespace and date format with TripleChecker.
- **[TripleChecker]** ERROR: VERY close match to “license.”
- **[TripleChecker]** Error loading - No parser available.
- **[WebVOWL]** ERROR "There is nothing to visualize.”
- **[OOPS]** OOPS Pitfall P36: URI contains file extension.
- **[OOPS]** OOPS Pitfall P37: Ontology not available on the Web.
- **[LODE]** Reason: An empty sequence is not allowed as the value of variable \$rdf.
- **[LODE]** Reason: A sequence of more than one item is not allowed as the @select attribute of xsl:sort.
- **[LODE]** Reason: org.xml.sax.SAXParseException; lineNumber: 1; columnNumber: 1; Content is not allowed in prolog.
- **[LODE]** The source can’t be downloaded in any permitted format. Connection reset Connection reset Connection reset Connection reset Connection reset Connection reset.
- **[LODE]** The source can’t be downloaded in any permitted format. Received fatal alert: protocol\_version.
- **[LOV]** Bad IRI: <http://example.com> Code: 57/REQUIRED\_COMPONENT\_MISSING in HOST: A component that is required by the schema is missing.

Some errors are more complicated to fix such as the server configuration. Also, when the syntax is checked with tools such as OWL Manchester validators, errors still arise when loading the ontologies.

## 5.2 Semantic Web Community Evaluation Criteria

To leverage and identify value for our PerfectO methodology, we discuss the evaluation criteria as outlined by the International Semantic Web Conference (ISWC) conference and their call for resources (e.g., ontologies) track papers [55].

**Impact Criteria.** PerfectO has a significant impact outside of the semantic web community. It has been mainly applied to the IoT community but is generic enough to be applied in other domains. As demonstrated in Section 4.2, PerfectO has been used with the LOV4IoT ontology catalog which covers more than 20 applicative domains (introduced in Section 2.1). It is the first innovative platform for semantic interoperability and disseminates best practices and existing tools as a set of catalogs following “Knowledge Directory Services” approaches. Our first step is to focus on ontology interoperability and improvement. Instead of saying that our tool is better than existing ones, we exploit their potential when they already provide web services. Our literature review shared in a structured way on the web through FAQs and mindmaps explained in Section 8.1 is also useful for teaching, tutorials, etc. We have in mind to have

a similar approach for datasets, reasoning, and querying, as explained in Section 4.1. We are aware that researchers can consider our methodology as not innovative since we reuse and integrate tools, but we are following software engineering research methodologies [15].

**Availability Criteria.** The PerfectO software is under the GNU GPLv3 license. A GitHub repository has been initiated <https://github.com/perfectkb/perfecto> which stands for perfect knowledge bases. The LOV4IoT catalog has been designed to extend the LOV ontology catalog since LOV inserts only new ontologies when best practices are followed. Due to this experience, we realized that the learning curve to improve ontologies and use the tools needs to be reduced and disseminated within other communities (e.g., IoT community). This work is aligned with our current activities at improving ontology quality in any projects designed at the Knoesis research center until 2020. PerfectO is hosted on Google Application Engine to avoid any server and DNS maintenance related issues.

**Reusability Criteria.** We provide a "Semantic Web Best Practices for Dummies" documentation [WR27] to improve ontologies. More than 200 ontology URLs have been referenced with the LOV4IoT dataset which has been integrated with PerfectO. PerfectO can be applied to any ontology URL. It provides online tools; no set up is required which reduces the development time. It can be extended with more tools to improve ontologies. For instance, we can integrate ontology matching tools such as LogMap, and ontology catalogs such as BioPortal and LOV to automatically suggest a new ontology. We are aware of the current limitations of PerfectO: some tools cannot load some ontologies. We try to understand why to help beginners solve the issues and report the solutions within the documentation.

**Design and Technical Quality Criteria.** We classify, reference, synthesize, vulgarize and disseminate best practices learned from the semantic web community to other communities (e.g., IoT). We reuse existing tools for automatic visualization, documentation and ontology quality as explained in Section 8.3.

### 5.3 Discussions and Lessons Learned

**Lessons Learned.** We shared the lessons learned to fix the issues to improve the ontologies with the tools suggested by PerfectO (see Table 3) in the "Semantic Web Best Practices for Dummies" documentation mentioned above. A summary is presented as a set of rules in Table 2, where we synthesize 16 rules to disseminate best practices. For each rule, we provide examples of bad practices and best practices to help beginners in their learning journey in a set of slides entitled "Step-by-step tutorial to improve the ontology quality, dissemination, reuse" [WR28]. This work is an enhancement of our previous work [19].

Rule Number	Description	Difficulty
Rule 1	Finding a good ontology name	*
Rule 2	Finding a good ontology namespace	**
Rule 3	Sharing your ontology online	**
Rule 4	Adding ontology metadata	**
Rule 5	Adding rdfs:label, rdfs:comment, dc:description for each concept and property	*
Rule 6	All classes start with an uppercase and properties with a lowercase.	*
Rule 7	Submitting your ontology to ontology catalogs	**
Rule 8	Reusing and linking ontologies	***
Rule 9	Deferenceable URI copy paste the namespace URL of your ontology in a web browser to get the code	**
Rule 10	Checking syntax validator	*
Rule 11	Adding ontology documentation	*
Rule 12	Adding ontology visualization	*
Rule 13	Improving Ontology Design	***
Rule 14	Improving dereferencing URI and content negotiation	***
Rule 15	Ontology can be loaded with ontology editors (e.g., Protege)	**
Rule 16	Registering your ontology on prefix catalogs	*

Table 2. Ontology best practices: check list summary

We are highly encouraging validation tools to provide REST APIs. However, REST APIs need to be maintained to ease PerfectO development. PerfectO’s main limitation is the usage of external web services which can be offline.

**Discussions.** Ontologies designed within projects (e.g., European projects) are more impactful, since they are used by different partners and for real use cases. For instance, the FIESTA-IoT ontology has been employed within the FIESTA-IoT H2020 EU project involving 14 academic and industrial partners. However, the major issue of European projects is the lack of ontology maintenance once the project is finished.

PerfectO improves ontology quality. For instance, in July 2019, best practices have been disseminated to the robotics community: IEEE P1872.2 Standard for Autonomous Robotics (AuR) ontology.

## 6 Conclusion and Future Work

Designing good quality ontologies is often neglected due to the lack of knowledge regarding the best practices. Getting familiar with ontology quality tools and best practices requires a steep learning curve and a lot of effort (e.g., for the IoT community). **PerfectO** assists ontology designers in this process towards improving ontologies to be reused in other projects. **PerfectO** selects and classifies a sub-set of tools providing an online interface or a web service simple to use. Those tools help to enhance ontologies and synthesize a set of practices. It

has a significant impact since it has been designed to be applied to the LOV4IoT ontology catalog which covers more than 20 application domains.

As future work, we plan to use the lessons learned from the evaluated ontologies to fix the ontology automatically and provide a ranking system to suggest the ontology fitting the ontology developer's needs.

## 7 Acknowledgments

This work has partially received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857237 (Interconnect), Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, Insight Centre for Data Analytics and H2020 FIESTA-IoT-CNECT-ICT-643943. The opinions expressed are those of the authors and do not reflect those of the sponsors.

## References

1. Paulheim, H.: Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. IOS Press Semantic web Journal (IF: 2.224 in 2018) (2017)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. International Journal on Semantic Web and Information Systems (2009)
3. Gruber, T.R.: Toward Principles for the Design of Ontologies used for Knowledge Sharing? International Journal of Human-Computer Studies (1995)
4. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Gayo, J.E.L., Kirrane, S., Neumaier, S., Polleres, A., et al.: Knowledge graphs. arXiv preprint arXiv:2003.02320 (2020)
5. Vandenbussche, P.Y., Atemezing, G.A., Poveda-Villalón, M., Vatant, B.: Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web. Semantic Web Journal (2016)
6. Gyrard, A., Zimmermann, A., Sheth, A.: Building IoT based applications for Smart Cities: How can ontology catalogs help? IEEE Internet of Things Journal (2018)
7. Gyrard, A., Bonnet, C., Boudaoud, K., M.Serrano: LOV4IoT: A second life for ontology-based domain knowledge to build Semantic Web of Things applications. In: IEEE International Conference on Future Internet of Things and Cloud. (2016)
8. Gyrard, A., Atemezing, G., Bonnet, C., Boudaoud, K., Serrano, M.: Reusing and Unifying Background Knowledge for Internet of Things with LOV4IoT. In: IEEE International Conference on Future Internet of Things and Cloud. (2016)
9. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality Assessment for Linked Data: A Survey. IOS Press Semantic Web Journal (IF: 2.224 in 2018) (2015)
10. McDaniel, M., Storey, V.C., Sugumaran, V.: Assessing the Quality of Domain Ontologies: Metrics and an Automated Ranking System. Data & Knowledge Engineering (IF: 1.583 in 2018) (2018)
11. Raad, J., Cruz, C.: A Survey on Ontology Evaluation Methods. In: KEOD. (2015)
12. Hloman, H., Stacey, D.: Approaches, Methods, Metrics, Measures, and Subjectivity in Ontology Evaluation: A Survey. Semantic Web Journal (2014)
13. García, J., Jose'García-Peñalvo, F., Therón, R.: A Survey on Ontology Metrics. In: World Summit on Knowledge Society, Springer (2010)

14. Fernández-López, M., Poveda-Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: Why are Ontologies not Reused Across the Same Domain? *Journal of Web Semantics* (2018)
15. Rus, I., Lindvall, M.: Knowledge Management in Software Engineering. *IEEE Software Journal* (IF: 2.879 in 2018) (2002)
16. Serrano, Martin and Barnaghi, Payam and Carrez, Francois and Cousin, Philippe and Vermesan, Ovidiu and Friess, Peter: Internet of Things IoT Semantic Interoperability: Research Challenges, Best Practices, Recommendations and Next Steps. Technical report, IERC AC4 (2015)
17. Agarwal, R., Fernandez, D.G., Elsaleh, T., Gyrard, A., Lanza, J., Sanchez, L., Georgantas, N., Issarny, V.: Unified IoT ontology to enable interoperability and federation of testbeds. In: *IEEE World Forum on Internet of Things*. (2016)
18. FIESTA IoT consortium, E.: FIESTA-IoT project Deliverable 6.1 Design of Global Market Confidence Programme on IoT interoperability (2016)
19. Gyrard, A., Serrano, M., Ateazing, G.: Semantic Web Methodologies, Best Practices and Ontology Engineering Applied to Internet of Things. In: *IEEE World Forum on Internet of Things*. (2015)
20. Suárez-Figueroa, M.C.: NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse. PhD thesis, Universidad Politecnica de Madrid, Facultad de Informatica, Departamento de Inteligencia Artificial (2010)
21. Noy, N.F., McGuinness, D.L.: *Ontology Development 101: A Guide to Creating your First Ontology* (2001)
22. Zaslavsky, A., Perera, C., Georgakopoulos, D.: Sensing as a Service and Big Data. arXiv preprint arXiv:1301.0159 (2013)
23. Gyrard, A., Serrano, M.: Connected Smart Cities: Interoperability with SEG 3.0 for the Internet of Things. In: *30th IEEE International Conference on Advanced Information Networking and Applications Workshops*. (2016)
24. Rezaei, R., Chiew, T.K., Lee, S.P., Aliee, Z.S.: Interoperability Evaluation Models: A Systematic Review. *Computers in Industry* (2014)
25. Serrano, Martin and Barnaghi, Payam and Cousin, Philippe: Semantic Interoperability: Research Challenges, Best Practices, Solutions and Next Steps, IERC AC4 Manifesto. Technical report, European Research Cluster on the Internet of Things, AC4 (2014)
26. Gyrard, A., Bonnet, C.: Semantic Web best practices: Semantic Web Guidelines for domain knowledge interoperability to build the Semantic Web of Things. OneM2M International Standard, Management, Abstraction and Semantics (MAS) Working Group 5, April 2014, Eurecom (2014)
27. Murdock, P., Bassbouss, L., Bauer, M., Alaya, M.B., Bhowmik, R., Brett, P., Chakraborty, R.N., Dadas, M., Davies, J., Diab, W., et al.: *Semantic Interoperability for the Web of Things* (2016)
28. Bauer, M., Baqa, H., Bilbao, S., Corchero, A., Daniele, L., Esnaola, I., Fernandez, I., Franberg, O., Garcia-Castro, R., Girod-Genet, M., Guillemin, P., Gyrard, A., Kaed, C.E., Kung, A., Lee, J., Lefrançois, M., Li, W., Raggett, D., Wetterwald, M.: *Semantic IoT Solutions - A Developer Perspective (Semantic Interoperability White Paper Part I)* (2019)
29. Bauer, M., Baqa, H., Bilbao, S., Corchero, A., Daniele, L., Esnaola, I., Fernandez, I., Franberg, O., Garcia-Castro, R., Girod-Genet, M., Guillemin, P., Gyrard, A., Kaed, C.E., Kung, A., Lee, J., Lefrançois, M., Li, W., Raggett, D., Wetterwald, M.: *Towards semantic interoperability standards based on ontologies (Semantic Interoperability White Paper Part II)* (2019)

30. Grüninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. (1995)
31. Ma, X., Fu, L., West, P., Fox, P.: Ontology Usability Scale: Context-Aware Metrics for the Effectiveness, Efficiency and Satisfaction of Ontology Uses. *Data Science Journal* (2018)
32. Corcho, O., Fernández-López, M., Gómez-Pérez, A.: Methodologies, Tools and Languages for Building Ontologies. Where is their Meeting Point? *Elsevier Data & knowledge engineering Journal* (IF: 1.583 in 2018) (2003)
33. Suarez-Figueroa, M.C., Gomez-Perez, A., Fernandez-Lopez, M.: The NeOn Methodology for Ontology Engineering. In: *Ontology engineering in a networked world*. Springer (2012)
34. Staab, S., Studer, R.: *Handbook on Ontologies*. Springer Science & Business Media (2013)
35. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: *Methontology: From Ontological Art Towards Ontological Engineering*. (1997)
36. Hitzler, P., Gangemi, A., Janowicz, K.: *Ontology Engineering with Ontology Design Patterns: Foundations and Applications*. IOS Press (2016)
37. Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: OOPS!(Ontology Pitfall Scanner!): An On-Line Tool for Ontology Evaluation. *International Journal on Semantic Web and Information Systems* (2014)
38. Duque-Ramos, A., Fernández-Breis, J.T., Iniesta, M., Dumontier, M., Aranguren, M.E., Schulz, S., Aussenac-Gilles, N., Stevens, R.: Evaluation of the OQuaRE Framework for Ontology Quality. *Expert Systems with Applications* (2013)
39. Duque-Ramos, A., Fernández-Breis, J.T., Stevens, R., Aussenac-Gilles, N.: OQuaRE: A SQuaRE-Based Approach for Evaluating the Quality of Ontologies. *Journal of Research and Practice in Information Technology* (H Index=21) (2011)
40. Fernández, M., Overbeeke, C., Sabou, M., Motta, E.: What Makes a Good Ontology? A Case-Study in Fine-Grained Knowledge Reuse. In: *Asian Conference on The Semantic Web*. Springer (2009)
41. Tartir, S., Arpinar, I.B.: Ontology Evaluation and Ranking Using OntoQA. In: *Semantic Computing, 2007. ICSC 2007. International Conference on*, IEEE (2007)
42. Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P., Aleman-Meza, B.: OntoQA: Metric-Based Ontology Quality Analysis. (2005)
43. Brank, J., Grobelnik, M., Mladenić, D.: A Survey of Ontology Evaluation Techniques. (2005)
44. Burton-Jones, A., Storey, V.C., Sugumaran, V., Ahluwalia, P.: A Semiotic Metrics Suite for Assessing the Quality of Ontologies. *Data & Knowledge Engineering* (2005)
45. Lozano-Tello, A., Gómez-Pérez, A.: OntoMetric: A Method to Choose the Appropriate Ontology. *Journal of Database Management* (2004)
46. Vrandečić, D.: Ontology Evaluation. In: *Handbook on Ontologies*. Springer (2009)
47. Vrandečić, D., Gangemi, A.: Unit tests for ontologies. In: *On the Move to Meaningful Internet Systems OTM Workshops*, Springer (2006)
48. Gangemi, A., Presutti, V.: *Ontology Design Patterns*. In: *Handbook on ontologies*. Springer (2009)
49. Bezerra, C., Freitas, F., Euzenat, J., Zimmermann, A.: ModOnto: A tool for modularizing ontologies. In: *Proc. 3rd workshop on ontologies and their applications (Wonto)*. (2008)
50. Garijo, D.: WIDOCO: a Wizard for Documenting Ontologies. In: *International Semantic Web Conference (ISWC, A-rank conference)*, Springer (2017)

51. Fielding, R.T., Taylor, R.N.: Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology (TOIT) (2002)
52. Kolbe, N., Kubler, S., Le Traon, Y.: Popularity-driven ontology ranking using qualitative features. In: International Semantic Web Conference, Springer (2019)
53. Olivares-Alarcos, A., Beßler, D., Khamis, A., Goncalves, P., Habib, M.K., Bermejo, J., Barreto, M., Diab, M., Rosell, J., Quintas, J., Olszewska, J., Nakawala, H., Pignaton, E., Gyrard, A., Borgo, S., Alenya, G., Beetz, M., Li, H.: A Review and Comparison of Ontology-based Approaches to Robot Autonomy. (2019)
54. Gyrard, A., Sheth, A.: IAMHAPPY: Towards An IoT Knowledge-Based Cross-Domain Well-Being Recommendation System for Everyday Happiness. (2019)
55. F, L.C., V, T.: ISWC 2017 Resources Track: Author and Reviewer Instructions (2017)
56. Buzan, T., Buzan, B.: The mind map book: How to use radiant thinking to maximize your brain's untapped potential. (1996)
57. McBride, B.: Jena: A semantic web toolkit. Internet Computing, IEEE (2002)
58. Tejo-Alonso, C., Berrueta, D., Polo, L., Fernández, S.: Metadata for Web Ontologies and Rules: Current Practices and Perspectives. In: Metadata and Semantic Research. Springer (2011)
59. Peroni, S., Shotton, D., Vitali, F.: Tools for the automatic generation of ontology documentation: A task-based evaluation. In: Computational Linguistics: Concepts, Methodologies, Tools, and Applications. IGI Global (2014)
60. Lohmann, S., Link, V., Marbach, E., Negru, S.: WebVOWL: Web-based visualization of ontologies. In: Knowledge Engineering and Knowledge Management. Springer (2014)
61. Berrueta, D., Fernández, S., Frade, I.: Cooking http content negotiation with vapour. In: 4th Workshop on Scripting for the Semantic Web (SFSW), Citeseer (2008)

## Web References

- [WR1] Mother IoT device: <https://sen.se/store/mother/>
- [WR2] Apple HealthKit:  
<http://bit.ly/2xBFo8x>
- [WR3] IoT Cisco's predictions: <http://bit.ly/2JqJLdj>
- [WR4] Google Knowledge Graph: <https://www.youtube.com/watch?v=mmQ16VGvX-c>
- [WR5] LOV4IoT: <http://lov4iot.appspot.com/>
- [WR6] Jena Framework Documentation: <https://jena.apache.org/>
- [WR7] Jena on GitHub: <https://github.com/apache/jena>
- [WR8a] Oops GUI: <http://oops.linkeddata.es/>
- [WR8b] Oops Web Service: <http://oops-ws.oeg-upm.net/>
- [WR9] Triple Checker GUI: <http://graphite.ecs.soton.ac.uk/checker/>
- [WR10] Triple Checker on GitHub: <https://github.com/cgutteridge/TripleChecker>
- [WR11] LOV GUI: <http://lov.okfn.org/dataset/lov/>
- [WR12] LOV Back End Java code on Github:  
<https://github.com/pyvandenbussche/lovScripts>
- [WR13] LOV JavaScript code for the GUI on GitHub:  
<https://github.com/pyvandenbussche/lov>
- [WR14] Parrot GUI: <http://ontorule-project.eu/parrot/parrot>
- [WR15] Parrot Java code on Bitbucket:

<https://bitbucket.org/fundacionctic/parrot/wiki/Home>  
 [WR16] LODÉ GUI: <http://www.essepuntato.it/lode>  
 [WR17] LODÉ Java code on GitHub: <https://github.com/essepuntato/LODE>  
 [WR18] WebVOWL GUI: <http://vowl.visualdataweb.org/webvowl.html>  
 [WR19] WebVOWL JavaScript code on GitHub:  
<https://github.com/VisualDataWeb/WebVOWL>  
 [WR20] Vapour GUI: <http://linkeddata.uriburner.com:8000/vapour>  
 [WR21] Vapour code on Bitbucket:  
<https://bitbucket.org/fundacionctic/vapour/wiki/Home>  
 [WR22] Vapour JavaScript API: <http://vapour.sourceforge.net/api/>  
 [WR23] OWL Manchester GUI: <http://visualdataweb.de/validator/>  
 [WR24] NeON ontology methodology: <http://neon-toolkit.org/>  
 [WR25] OQuaRE ontology quality tool: <http://miuras.inf.um.es:9080/oqmodelsliteclient/>  
 [WR26] Linked Data blog: <http://linkeddata.org/home>  
 [WR27] Semantic Web Best Practices for Dummies Documentation:  
<http://bit.ly/2XB9jsa>  
 [WR28] Slides step-by-step tutorial to improve the ontology quality, dissemination, reuse, etc. Semantic Web Best Practices:  
<https://goo.gl/Rg4cGr>  
 [WR29] Domain Ontology Ranking System (DoORS) prototype:  
<https://owlparser.herokuapp.com/>  
 [WR30] Ontology Design Patterns (ODPs) wiki:  
<http://ontologydesignpatterns.org/>

## 8 Appendix

### 8.1 Catalogs of Tools

The PerfectO web site provides the *Catalogs of tools* menu (depicted in Figure 8) to easily give access to software URLs and publications. The *Catalogs of tools* references tools for (1) ontology improvement which is the focus of this paper, (2) dataset quality, (3) querying, and (4) reasoning. Figure 9 demonstrates an extensive literature survey within the table of contents classified in two different ways:

- A bullet list referencing tool’s URLs and scientific publications are displayed simply with clickable links (see Figure 6 as an example).
- Mind maps [56] are recognized as a useful methodology and a powerful graphics technique used to translate what’s in the mind into a visual picture. Since mind mapping works like the brain, it allows us to organize and understand information faster and better. We designed mind maps to answer Frequently Asked Questions (FAQs). Figure 11 shows the Mind maps for ontology catalogs. Figure 12 illustrates the Mind maps for ontology methodologies. Figure 13 illustrates the Mind maps for ontology validation.

The catalogs of tools are available online<sup>19</sup>; it covers numerous research domains: 1) ontology documentation, 2) ontology catalogs and semantic search engines, 3) ontology methodologies, 4) ontology validators, 5) ontology validators for IoT, 6) ontology visualization, 7) collaborative vocabulary development, 8) ontology evaluation, and 9)

<sup>19</sup> [http://perfectsemanticweb.appspot.com/?p=ontology\\_sota](http://perfectsemanticweb.appspot.com/?p=ontology_sota)

ontology repair. Extensive work has been done in ontology documentation (not covered in the Related Work Section) and structured within a mind map, but referenced by the *Catalogs of Tools* as displayed in Figure 10. We guide ontology designers with LODE and Parrot tool since web services are provided, as shown in Section 8.3 and Table 3. We cover some of those topics (ontology methodology, ontology evaluation, ontology metrics, quality, and relevant tools) in Section 3. Ontology designers can contribute to enrich the set of *Catalogue of Tools* by using a Google Form interface<sup>20</sup>.

### Collaborative Vocabulary Development

- [WebProtege] Paper: [A Framework for Ontology Evolution in Collaborative Environments](#) [Noy et al. ISWC 2016] (Change and Annotation Ontology (CHAO) ontology)
- [WebProtege] Paper: [Web-Protege: A lightweight OWL Ontology Editor for the Web](#) [Tudorache et al. 2008]
- [WebProtege] Paper: [Supporting collaborative ontology development in Protege](#) [Tudorache et al. 2008]
- [WebProtege] Tool: [WebProtege Tool](#)
- [VoCol] Paper: [VoCol: An agile methodology and environment for collaborative vocabulary development](#) [Petersen et al. 2015]
- [VoCol] Paper: [An integrated environment to support version-controlled vocabulary development](#) [Halilaj et al. EKAW 2016]
- [VoCol] [Code on GitHub](#)
- [Simperl et al.] [Collaborative ontology engineering: a survey](#) [Simperl et al. 2014]
- [Git4Voc] Paper: [Git4Voc: Git-based Versioning for Collaborative Vocabulary Development](#) [Halilaj et al. 2016]
- [Git4Voc] [Git Best Practices](#)
- [Moki] Paper: [Modeling in a Wiki with MoKi: Reference Architecture, Implementation, and Usages](#) [Ghidini et al. 2012]
- [Moki] Paper: [MoKi: a Wiki-Based Conceptual Modeling Tool](#) [Ghidini et al. 2010]
- [Moki] Tool: [Moki tool](#)

Fig. 6. Literature survey for collaborative vocabulary development with clickable links

## 8.2 Dr. PerfectO Availability of Tools (DPAT)

DPAT tool (depicted in Figure 7 and introduced in Section 4.1), online [http://perfectsemanticweb.appspot.com/?p=availability\\_tools](http://perfectsemanticweb.appspot.com/?p=availability_tools), checks the availability of reusable tools in case the server is down. Each row provides: (1) the name of the tool, (2) its usage, (3) the clickable tool's URL, and (4) the tool availability (displayed as images: OK or NOT OKAY). For instance, the LODE ontology documentation web service runs well when DPAT has been deployed.

## 8.3 PerfectO Guidance: The Most Accessible Tools for Ontology Engineering

The learning curve for software engineering can be extremely high when developers are not familiar with the same programming languages and libraries used to build the tools. A set of tools that can be considered <easy-to-use> if: (1) The tools provide GUIs and web services, (2) the documentation is available and well-explained, (3) the ontologies can be evaluated with tools offering diverse functionalities, and (4) software setup configuration is not required.

The classification of tools that we have selected is available within Table 3 to support the *Ontology Improvement* methodology (explained in Section 4.2). The table

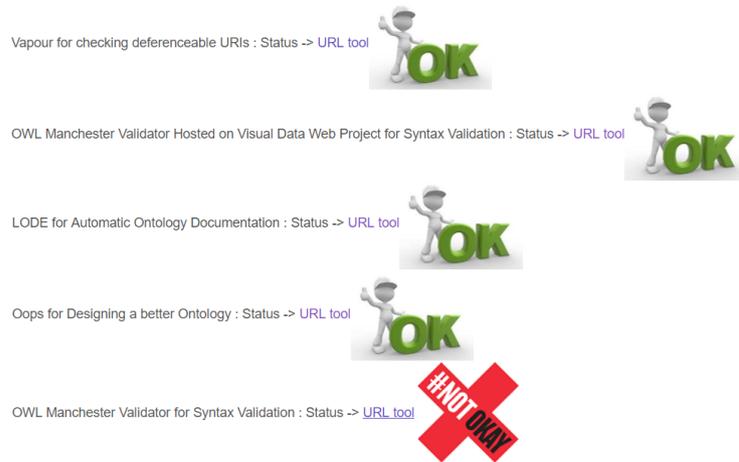
<sup>20</sup> <http://perfectsemanticweb.appspot.com/?p=updateCatalogueForm>

Tool Name and Publication	Validation Requirement	GUI	Web Service	Code Availability	Maintained
Jena [57]	Serialization Syntactic	[WR6]	✓ API	✓ GitHub Java code [WR7]	✓ High
Oops [37]	Ontology Consistency	[WR8a]	✓ [WR8b]	✗	✓ High
Triple Checker	Syntactic	[WR9]	✓	✓ GitHub PHP code [WR10]	✓ High
LOV [5]	Discoverability	[WR11]	✓	✓ Github Back end - Java [WR12] GUI - Javascript [WR13]	✓ High
Parrot [58]	Documentation	[WR14]	✓	✓ Bitbucket Java [WR15]	✓ Hosted by Mondeca now
LODE [59]	Documentation	[WR16]	✓	✓ Java on GitHub [WR17]	✓
WebVOWL [60]	Visualization	[WR18]	✓	✓ GitHub JavaScript [WR19]	✓ High
Vapour [61]	Discoverability	[WR20]	✓ Java JavaScript API [WR22]	✓ [WR21]	✓ Medium
OWL Manchester	Syntactic	[WR23]	✗	✗	✓ Medium

**Table 3.** Reusable tools for the ontology improvement

is a way to organize the multiple tested technologies and if there is an available source for documenting it. For instance, WebVOWL tool can be used to provide automatic ontology graph visualization, Parrot for automatic documentation, etc. In the table, the first column is dedicated to the tool name, and scientific publication is available. The second column explains the requirement satisfied. The third column provides the GUI interface URL. The fourth column indicates the web service or API if available. The fifth column contains the code URL if accessible. The sixth column explains the maintainability of the tools. The web services are more convenient to integrate when developing the methodology, but the implementation depends on web reliability and the maintenance of web services. Sometimes the servers hosting the web services are down, or when new versions are released, it has an impact on the implementation. When the tools are open source, such dependencies are avoided, but it is more time-consuming for developers to get into the code based on various languages and technologies. It is another reason demonstrating the needs to help ontology designers. In Table 3, within the maintained column: *High* means that the community behind the tools is reactive when issues arise such as server down, fixing bugs, answering questions or adding new functionalities. *Medium* means that the tools is frequently down, due to server issues.

More tools will be integrated later since we are facing the issues of the availability of tools as well. For this reason, a parallel work was to develop the *Dr. PerfectO Availability of Tools (DPAT)* component demonstrator (introduced in Section ).



**Fig. 7.** Dr. PerfectO Availability of Tools (DPAT)



**Fig. 8.** Catalogs of Tools

## FAQ for Ontologies: Exploring the State of the Art with Interactive MindMaps

- [FAQ: How to document ontologies? \(Interactive Mindmap\)](#)
- [FAQ: How to find ontologies? With Ontology Catalogues, etc. \(Interactive Mindmap\)](#)
- [FAQ: What are the ontology Methodologies? \(Interactive Mindmap\)](#)
- [FAQ: How to validate ontologies? \(Interactive Mindmap\).](#)
- [FAQ: Ontology and Dataset Validators for IoT? \(Interactive Mindmap\)](#)
- [FAQ: How to visualize ontologies? \(e.g., WebVOWL\) are tools to visualize ontologies.](#)
- [FAQ: Additionnal Recommendations!](#)
- [SOTA: Collaborative Vocabulary Development](#)
- [SOTA: Ontology Evaluation](#)
- [SOTA: Ontology Repair](#)
- [SOTA: Other References](#)

**Fig. 9.** State of the Art classifying ontology improvement tools

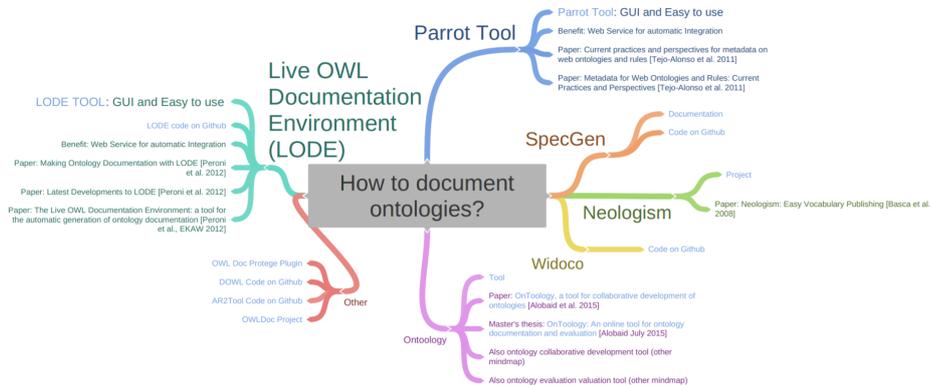


Fig. 10. Mind map classifying ontology documentation tools

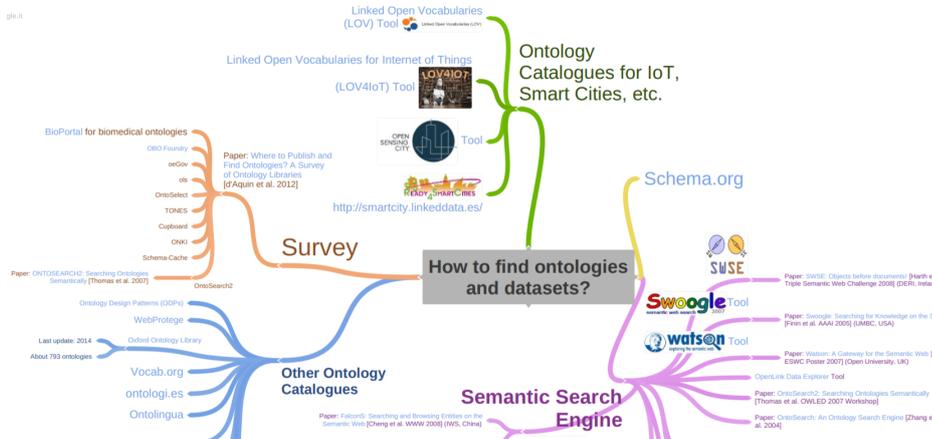


Fig. 11. Mind map classifying ontology catalog tools

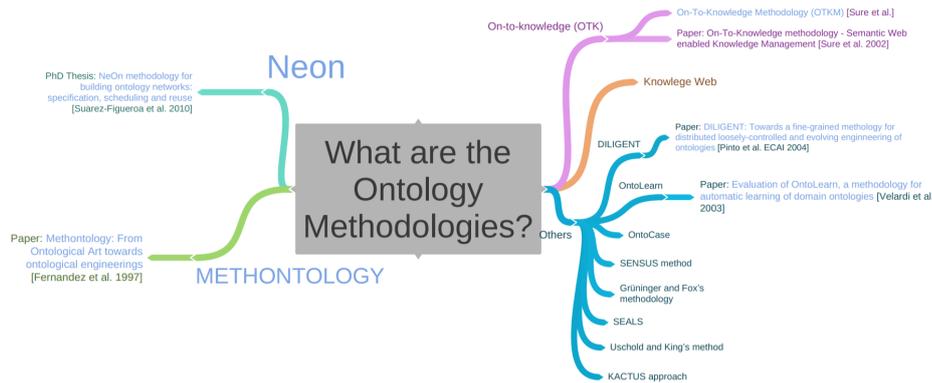


Fig. 12. Mind map classifying ontology methodologies



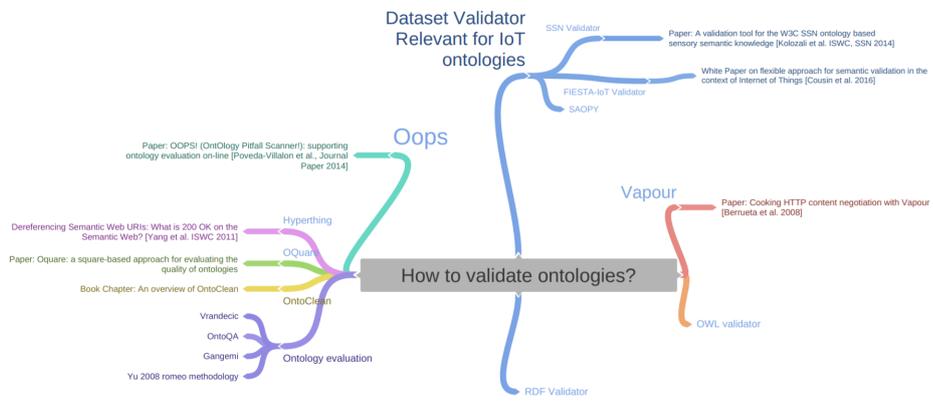


Fig. 13. Mind map classifying ontology validation tools