

SemWeb Best Practices for Dummies with PerfectO

Document Title	Title V2: SemWeb Best Practices for Dummies with PerfectO Title V1: SemWeb Best Practices in IoT for Dummies with PerfectO
Documentation URL	http://perfectsemanticweb.appspot.com/documentation/SemanticWebBestPracticesForDummies.pdf
Project	PerfectO PerfectO Web site: http://perfectsemanticweb.appspot.com/  <i>PerfectO references, classifies and provides tools to encourage Semantic Web Best Practices to achieve Semantic Interoperability by focusing on ontology improvement.</i>
Creator	Amelie Gyrard (Kno.e.sis - Ohio Center of Excellence in Knowledge-enabled Computing Wright State University, Ohio, USA) Previously (Ecole des Mines de Saint-Etienne, France)  amelie@knoesis.org
Contributors / Feedback to improve the documentation	Ghislain Atemezing (Mondeca, & Linked Open Vocabularies community, France) Martin Serrano (Insight Center for Data Analytics, NUIG, Galway, Ireland) Utkarshani Jaimini (Kno.e.sis, Wright State University, USA)
Last Updated	April 2019 • Test ontology matching with the LogMap web service. IF HTTPS URL, it seems it does not work. September 2018: • Step by step tutorial for improving Knoesis ontologies, slides done in parallel ◦ Semantic Web Best Practices • Updates rule 10 to 16 ◦ Prefix.cc, yasgui ontology namespace auto-completion

	August 2018: • Add more rules – checklist summary April 2018 • Table of content reorganization • Semantic Web Tutorials – Useful links • Tutorial for PURL • Various improvements – ontology best practices checklist summary March 2018 • Screenshot example to avoid bad practices • Improve check list summary • Various updates February 2018 • Adding Ontology metadata code within your ontology • Add useful reference to explain the difference between taxonomy, vocabulary, etc. • Improving ontology quality at Knoesis, share all best practices, tips and basics to be familiar with best practices. September 2017 • Issue with ontologies when suggest them to LOV August 2017 • Frequently seen bad practices section May 2017: • All errors encountered when evaluating various ontologies with the different tools.
Created	April 2017
Status	Ongoing Work
Goal	• Sharing a set of semantic best practices • Help you improve your ontologies • Documentation to help fix errors encountered to design perfect ontologies with existing Semantic Web (SemWeb) validation or documentation tools! ○ TripleChecker ○ OOPS ○ WebVOWL ○ Parrot ○ Vapour ○ LODE • Encouraging semantic interoperability.
Guidance Book	Dummies term is used for fun to follow the books! Do not take it personally! ☺

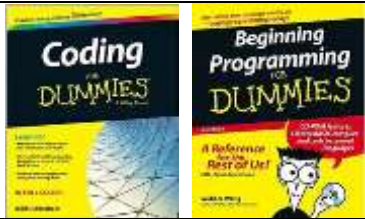
	
Acknowledgments	• Knoesis colleagues with fruitful discussions to improve this guide (Farahnaz Golroo, Shruti Kar, etc.). • The Linked Open Vocabularies (LOV) Team for sharing their knowledge regarding validation tools and best practices. ○ Pierre-Yves Vandenbussche ○ Maria Poveda (she also created the OOPS validation tool) ○ Bernart Vatan • Christopher Gutteridge (for his help with TripleChecker) • Steffen Lohmann (for his help with WebVOWL) • Carlos Tejo and Sergio Fernández (for their help with Parrot) • Laurens Rietveld (YASGUI SPARQL editor) • Paul Murdock for the idea of “Guide for dummies” and the Semantic Interoperability for the Web of Things [Murdock et al. 2016] White Paper

Table of Content

I.	Please cite our research work if you are using the PerfectO web site, this guide, etc.	7
II.	Semantic Web and Ontology Basics (UPDATED April 2018)	7
1.	Semantic Web Basics (CREATED April 2018)	7
2.	Designing your first ontology? Ontology Tutorials (UPDATED April 2018)	8
3.	Definitions (NEW February 2018).....	9
III.	Summary of some awesome tools to use	9
IV.	Ontology Best Practices - Checklist (NEW February 2018)	11
1.	Ontology Best Practices - Checklist Summary (UPDATED August 2018).....	11
2.	Rule 1: Finding a good ontology name (UPDATED March 2018)	12
1.	Rule 2: Finding a good ontology namespace (UPDATED April 2018)	13
a)	Permanent URL with PURL	13
b)	Permanent URL with W3id	14
2.	Rule 3: Sharing your ontology online (NEW February 2018)	14
3.	Rule 4: Adding ontology metadata (UPDATED March 2018, September 2018).....	15
4.	Rule 5: Adding rdfs:label, rdfs:comment, dc:description for each concept and property (UPDATED March 2018)	16
5.	Rule 6: All classes should start with an Uppercase. The properties should start with a lowercase (TO DO NEW February 2018)	17
6.	Rule 7: Submitting your ontology to ontology catalogs (NEW February 2018)	17
7.	Rule 8: Reusing and linking your ontology to other ontologies (TO DO - NEW February 2018).....	18
8.	Rule 9: Deferenceable URI: copy paste the namespace URL of your ontology in a web browser to get the code	19
9.	Rule 10 (TO DO).....	21
10.	Rule 11 (TO DO)	21
11.	Rule 12 (TO DO)	21
12.	Rule 13 (TO DO)	21
13.	Rule 14 (TO DO)	21
14.	Rule 15 (TO DO)	21
15.	Rule 16: Registering your ontology on prefix catalogs	21
V.	How to check the design of your ontology (TO DO - NEW February 2018)	24
VI.	Set of Tools to improve ontologies	24
1.	Ontology Documentation with Parrot.....	24
a)	Parrot Error: Unable to read input document: invalid mimeType "application/octet-stream" (returned by URI) for parrot	24
b)	Parrot Error: The ontology can be loaded but nothing is displayed	24
c)	Parrot Error: I/O Error: Server returned HTTP response code: 403 for URL	25

d) Parrot Error: Unable to read input document: application/rdf+xml parse error: Content is not allowed in prolog.	26
2. Syntax Validator with TripleChecker	27
a) TripleChecker Error: Possible match to "date"	27
b) TripleChecker Error: VERY close match to "license"	27
c) TripleChecker Error - loading- No parser available	28
3. Ontology Visualization with WebVOWL	28
a) WebVOWL Error: There is nothing to visualize	28
4. Improving Ontology Design with OOPS.....	28
a) OOPS Error: Pitfall 36 - URI contains file extension.....	29
b) OOPS Error: Pitfall 37 - Ontology not available on the Web	29
1. Ontology Matching with LogMap web service (NEW: April 2019).....	31
2. Ontology Documentation with LOD	31
a) LOD Error - Reason: An empty sequence is not allowed as the value of variable \$rdf.....	31
b) LOD Error - Reason: A sequence of more than one item is not allowed as the @select attribute of xsl:sort.....	31
c) LOD Error - Reason: org.xml.sax.SAXParseException; lineNumber: 1; columnNumber: 1; Content is not allowed in prolog.	32
d) LOD Error - The source can't be downloaded in any permitted format. # Connection reset # Connection reset # Connection reset # Connection reset # Connection reset.....	32
e) LOD Error - The source can't be downloaded in any permitted format. # Received fatal alert: protocol_version	32
VII. Suggesting Ontologies on LOV	33
VIII. Frequently Seen Bad Practices	33
IX. Useful Links.....	33
X. References.....	33

Table of figures

FIGURE 1. GETTING STARTED WITH APACHE JENA	8
FIGURE 2. WIMMICS TEAM, SEMANTIC WEB COURSE EXAMPLE	8
FIGURE 3. BAD EXAMPLES: YOU NEED TO FIND AN EXPLICIT NAME TO DESCRIBE YOUR ONTOLOGY	12
FIGURE 4. RENAME YOUR ONTOLOGY UNDER PROTÉGÉ (AND THINK ABOUT THE ONTOLOGY URL/NAMESPACE)	13
FIGURE 5. CREATE A NEW URL UNDER PURL (E.G., IOT/VOCAB/NAME-ONTO#)	13
FIGURE 6. LINK THE PURL URLS TO YOUR ONTOLOGY OR ANY WEB SITE	14
FIGURE 7. ADDING ONTOLOGY METADATA CODE WITHIN YOUR ONTOLOGY	15
FIGURE 8. BAD PRACTICE EXAMPLE: THERE IS A NEED TO DESCRIBE CONCEPTS OR PROPERTIES TO KNOW THEIR MEANING AND DEFINITION	17
FIGURE 9. GOOD PRACTICE: ADD RDFS:LABEL AND RDFS:COMMENT TO BETTER DESCRIBE CONCEPTS OR PROPERTIES	17
FIGURE 10. ONTOLOGY CATALOGUE MINDMAP WITH A SET OF TOOLS TO USE	18
FIGURE 11. BAD EXAMPLE: COPY PASTE THE URL NAMESPACE OF YOUR ONTOLOGY IN A WEB BROWSER (E.G., CHROME), YOU SHOULD NOT GET THE “404 – NOT FOUND” ERROR, BUT GET THE ONTOLOGY CODE OR THE DOCUMENTATION.	19
FIGURE 12. GOOD EXAMPLE: COPY AND PASTE AN ONTOLOGY NAMESPACE URL IN A WEB BROWSER TO GET THE CODE	20
FIGURE 13. UNABLE TO READ INPUT DOCUMENT: INVALID MIMEType "APPLICATION/OCTET-STREAM" (RETURNED BY URI) FOR PARROT	24
FIGURE 14. NO ERROR ENCOUNTERED WHEN LOADED THE ONTOLOGY BUT NOTHING IS DISPLAYED	25
FIGURE 15. I/O ERROR: SERVER RETURNED HTTP RESPONSE CODE: 403 FOR URL	25
FIGURE 16. UNABLE TO READ INPUT DOCUMENT: APPLICATION/RDF+XML PARSE ERROR: CONTENT IS NOT ALLOWED IN PROLOG	26
FIGURE 17. MISUSE OF TERMS FROM DUBLIN CORE NAMESPACE AND DATE FORMAT WITH TRIPLECHECKER	27
FIGURE 18. ERROR: VERY CLOSE MATCH TO “LICENSE”	27
FIGURE 19. ERROR LOADING- NO PARSER AVAILABLE	28
FIGURE 20. ERROR "THERE IS NOTHING TO VISUALIZE”	28
FIGURE 21. OOPS REPORT WITH CRITICAL PITFALLS	29
FIGURE 22. OOPS PITFALL 36 URI CONTAINS FILE EXTENSION	29
FIGURE 23. OOPS ERROR: P37: ONTOLOGY NOT AVAILABLE ON THE WEB	30
FIGURE 24. REASON: AN EMPTY SEQUENCE IS NOT ALLOWED AS THE VALUE OF VARIABLE \$RDF	31
FIGURE 25. LODE ERROR - REASON: A SEQUENCE OF MORE THAN ONE ITEM IS NOT ALLOWED AS THE @SELECT ATTRIBUTE OF XSL:SORT	31
FIGURE 26. REASON: ORG.XML.SAX.SAXPARSEEXCEPTION; LINENUMBER: 1; COLUMNNUMBER: 1; CONTENT IS NOT ALLOWED IN PROLOG.	32
FIGURE 27. LODE ERROR - THE SOURCE CAN'T BE DOWNLOADED IN ANY PERMITTED FORMAT. # CONNECTION RESET # CONNECTION RESET # CONNECTION RESET # CONNECTION RESET # CONNECTION RESET # CONNECTION RESET	32
FIGURE 28. THE SOURCE CAN'T BE DOWNLOADED IN ANY PERMITTED FORMAT. # RECEIVED FATAL ALERT: PROTOCOL_VERSION	32
FIGURE 29. LOV SUGGEST ERROR – SUGGESTING AN ONTOLOGY TO LOV	33

I. Please cite our research work if you are using the PerfectO web site, this guide, etc.

Please do not forget to cite our research work if you are using this documentation and the PerfectO web site. Thank you very much in advance.

Publications:

- [Semantic Web Methodologies, Best Practices and Ontology Engineering Applied to Internet of Things.](#)
 - IEEE World Forum on Internet of Things (WF-IoT), Milan, Italy, December 14-16, 2015
 - Amelie Gyrard, Martin Serrano, Ghislain Atemezang.
- [A survey and analysis of ontology-based software tools for semantic interoperability in IoT and WoT landscapes.](#)
 - IEEE 4th World Forum on Internet of Things (WF-IoT), 2018
 - Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet
- [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, Amelie Gyrard, Christian Bonnet.

II. Semantic Web and Ontology Basics (UPDATED April 2018)

This section recommends useful links. There are already numerous tutorials and courses on the web. We just provide few recommendations.

1. Semantic Web Basics (CREATED April 2018)

There are already nice tutorials to learn Semantic Web basics.

The [Jena Semantic Web Framework](#). Jena is a framework for Java developers.

They provide excellent tutorial to learn the basics¹ (see Figure 1):

- Jena tutorial - [An Introduction to RDF and the Jena RDF API](#)
- Jena tutorial - [SPARQL language, a SQL-like language to query semantic web data, called triples.](#)
- Jena tutorial - [Manipulating SPARQL using ARQ](#)
- OWL and ontologies to model and structure data
- Inference engine and rules

¹ https://jena.apache.org/getting_started/index.html

Getting started with Apache Jena

Apache Jena (or Jena in short) is a free and open source Java framework for building [semantic web](#) and [Linked Data](#) applications. The framework is composed of different APIs interacting together to process RDF data. If you are new here, you might want to get started by following one of the [tutorials](#). You can also browse the [documentation](#) if you are interested in a particular topic.

✈️ Tutorials

- [RDF API tutorial](#) - you will learn the essence of the semantic web and the graph representation behind RDF.
- [SPARQL tutorial](#) - will guide you to formulate expressive queries over RDF data.
- [Ontology API](#) - illustrates the usage of advanced semantic web features such as reasoning over your data using OWL.
- Finally, some of the tutorials are also available in Traditional Chinese, Portuguese and French.

Figure 1. Getting started with Apache Jena

We also recommend the [Semantic Web class from the Wimmics team](#), INRIA Sophia Antipolis, France (see Figure 2):

- [Corese](#) tutorial and tool

2013-2014

- [Corese](#)
- [Inference Rules](#)
- [TP Inference Rules](#)
- [Template Inference Rules](#)

2012-2013

- [Corese API](#)
- [Corese](#)
- [Corese Rules](#)
- [RIF](#)
- [TP Rules](#)
- [SPARQL 1.0](#)
- [TD SPARQL 1.0](#)
- [SPARQL 1.1](#)
- [SPARQL Update](#)
- [TD SPARQL 1.1](#)

Figure 2. Wimmics team, Semantic Web course example

More and more universities are having MOOC and Semantic Web course.

2. Designing your first ontology? Ontology Tutorials (UPDATED April 2018)

Enclosed a set of useful links:

- [Protégé](#): A software having a Graphical User Interface (GUI) to design and develop ontologies.
 - [Protégé Tutorial](#) [Horridge et al. 2011] – Design the Pizza ontology. Check if there is a more recent documentation.

- [101 Ontology Development methodology](#) [Noy et al. 2001] - Learn with the wine ontology and discover ontology best practices.
- The [Neon methodology](#) to discover more about ontology best practices.

Book Recommendation: Handbook of Ontologies [Staab et al. 2010]

3. Definitions (NEW February 2018)

What are the differences between **Taxonomy**, **Ontology**, **Thesaurus**, **Vocabulary**, **Ontology/namespace**, **Schema**, etc.?

	You can find those answers here: • Taxonomies & Controlled Vocabularies SIG explains the difference between Vocabularies, Taxonomies, Thesauri and Ontologies. • http://www.dataversity.net/taxonomy-vs-ontology-machine-learning-breakthroughs/
---	--

III. Summary of some awesome tools to use

Name Tool	Goal	URL	Reference
TripleChecker	Syntax validation	http://graphite.ecs.soton.ac.uk/checker/	No
OOPS	Design a better ontology	http://oops.linkeddata.es/	[3]
WebVOWL	Automatic Ontology Graph Visualization	http://visualdataweb.de/webvowl/#iri=http://xmlns.com/foaf/0.1/	[5]
Parrot	Automatic Ontology Documentation	http://ontorule-project.eu/parrot/parrot	[4]
LODE	Automatic Ontology Documentation	http://www.essepuntato.it/lode	[9]
Vapour	Check Deferencable URIs	http://linkeddata.uriburner.com:8000/vapour	[6]
OWL Manchester	Syntax validation	http://mowl-power.cs.man.ac.uk:8080/validator/ or http://visualdataweb.de/validator/	Not found yet
LOV Suggest	Suggest the ontology to LOV	http://lov.okfn.org/dataset/lov/suggest	[1]
PURL	Permanent URLs	https://archive.org/services/purl/	Not found yet
W3id	Permanent URLs	https://w3id.org/	

IV. Ontology Best Practices - Checklist (NEW February 2018)

Some recommendations to guide you to design a better ontology. Check all of those items.

1. Ontology Best Practices - Checklist Summary (UPDATED August 2018)

Rules	Description	More explanations	Difficulty (*, **, ***)	Status (Done, In progress) ?	Estimated time
Rule 1	Finding a good ontology name	Find an explicit name for your ontology, we frequently see “unnamed.owl”!	*		
Rule 2	Finding a good ontology namespace	Ideally on PURL or W3id. Otherwise, think about the server hosting the ontology (e.g., http://knoesis.org/ontology/nameOntology#)	**		15 mins
Rule 3	Sharing your ontology online	Accessible with an URL (http://knoesis.org/ontology/nameOntology#) No server? Push the ontology code on Github? No server? we can host the ontology code on the LOV4IoT server (but is the ontology stable version?)	**		
Rule 4	Adding ontology metadata	This is important to later reference the ontology on ontology catalogs, or even to provide automatic ontology visualization, ontology documentation, etc.	*		
Rule 5	Adding <code>rdfs:label</code> , <code>rdfs:comment</code> , <code>dc:description</code> for each concept and property	This is important to later provide automatic ontology visualization, documentation, etc. Some tools prefer <code>dc:description</code> , check which ones (e.g., LODE?).	*		
Rule 6	All classes should start with an uppercase and properties with a lowercase.	To follow usual software and ontology development guidelines.	*		
Rule 7	Submitting your ontology to ontology catalogs	Ontology catalogs: LOV, LOV4IoT, BioPortal. It depends on your applicative domain.	**		
Rule 8	Reusing and linking ontologies	Reuse an existing concepts and properties from an existing ontology/namespace (e.g., <code>ssn:Device</code>) Otherwise add <code>owl:EquivalentClass</code> , <code>owl:sameAs</code> , <code>owl:equivalentProperty</code> , etc.	***		

Rule 9	Deferenceable URI: copy paste the namespace URL of your ontology in a web browser to get the code	Important to automatize the tasks to automatically retrieve the ontology code for automatic analysis of ontologies	**		
Rule 10	Checking syntax validator	TripleChecker tool is an easy web service to use. It can check incorrect use of ontologies. Other tools: OWL Validator, RDF Validator, etc.	*		
Rule 11	Adding ontology documentation	Ontology documentation can be done automatically with easy to use tools by using their web services if you have labels and comments. E.g., LOD2, Widoco, Parrot.	*		
Rule 12	Adding ontology visualization	Usage of the WebVOWL tool to provide the ontology visualization automatically.	*		
Rule 13	Improving Ontology Design	Usage of the OOPS tool to improve the ontology design.	***		
Rule 14	Improving deferencing URI and content negotiation	Usage of the Vapour tool	***		
Rule 15	Loading the ontology on the most popular ontology editor	Should be able to be loaded under Protégé Ontology editor since it is a popular editor. Sometimes ontologies cannot be loaded or are empty.	**		
Rule 16	Registering your ontology on prefix catalogs	Registering your ontology (namespace and prefix) on prefix.cc. For instance, it is relevant for YASGUI auto-completion. Permanent URL are important (e.g., possible with PURL)	*		

2. Rule 1: Finding a good ontology name (UPDATED March 2018)

Do not forget to find a good name for your ontology. Frequently we find “unnamed.owl”, “schema.ttl” or “ontology.owl”! Please find an explicit name related to your project, etc.

Further, find the best prefix (.xml, .rdf, .rdfs, .owl)

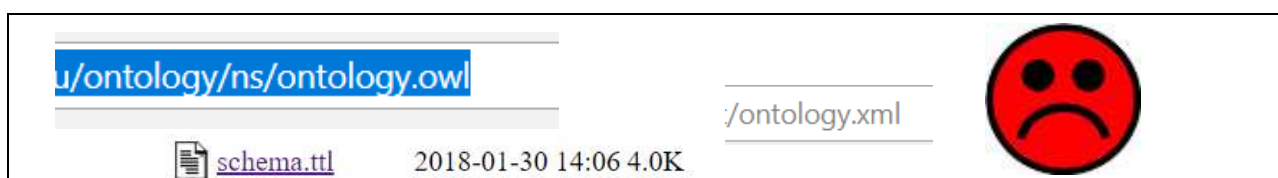


Figure 3. Bad examples: You need to find an explicit name to describe your ontology

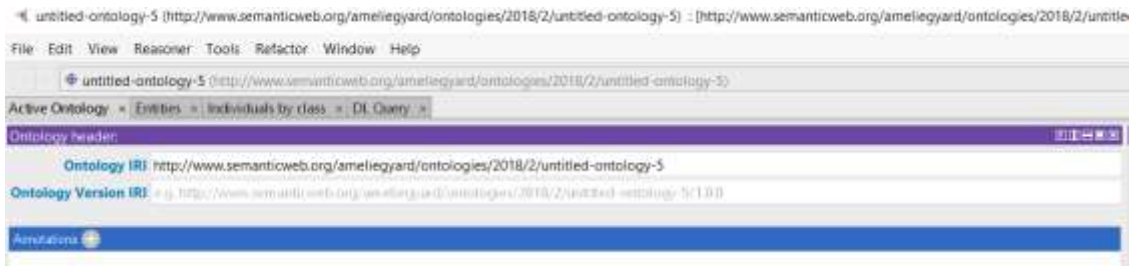


Figure 4. Rename you ontology under Protégé (and think about the ontology URL/namespace)

1. Rule 2: Finding a good ontology namespace (UPDATED April 2018)

When you copy paste the ontology URL or even the ontology namespace in a web browser you should get the ontology documentation or the ontology code and not Error 404, page not found. 😊

a) Permanent URL with PURL

Get a permanent URL with the PURL tool: <https://archive.org/services/purl/>

You can easily create a permanent URL with Purl. It takes less than 15 minutes:

- Create an account and register under this tool.
- Create a new domain.
 - We have created <http://purl.org/iot/ontology/> for any IoT ontologies
 - Example: <http://purl.org/iot/ontology/fiesta-iot#>
 - Example: <http://purl.org/iot/vocab/m3-lite#>

PURL Administration

Home

PURLs are persistent URLs, they provide permanent addresses for resources on the web.

Search

Create a new domain

Figure 5. Create a new URL under PURL (e.g., [iot/vocab/name-onto#](http://purl.org/iot/vocab/name-onto#))

new PURL '/perfecto' saved

Domain: /perfecto

[Home](#) / [Domain: /perfecto](#)

[view subdomains](#)

PURLs

name	type	target	created
/perfecto	302 Found	http://perfectsemanticweb.appspot.com/	2018-04-02 20:25:19

Add a new PURL to this domain

name type target URL

example: /perfecto/duck example: http://example.org/

Figure 6. Link the PURL URLs to your ontology or any web site

Example:

new PURL '/dao' saved

Domain: /dao

[Home](#) / [Domain: /dao](#)

[view subdomains](#)

PURLs

name	type	target	created
/dao	302 Found	http://knoesis.org/ontology/DAO.owl	2018-08-31 16:27:22

Add a new PURL to this domain

name type target URL

example: /dao/duck example: http://example.org/

b) [Permanent URL with W3id](#)

You can do something similar with the w3id tool: <https://w3id.org/>

However, we have noticed that your URLs will start with HTTPS which might generate issues with other tools hosted on non-secured servers!

For this reason, we encourage PURL also because it is faster to create a PURL URL.

[2. Rule 3: Sharing your ontology online \(NEW February 2018\)](#)

Ideally, share your ontology code online, for instance you can either:

- Share the file on a server
- Share the file on Github, BitBucket, etc.

3. Rule 4: Adding ontology metadata (UPDATED March 2018, September 2018)

Adding ontology metadata is important. We need to know when the ontology has been created, if the ontology is still maintained, the creators to contact them, the related papers explaining more about the project and the ontology, etc.

	Why is it important? Adding ontology metadata is important to be referenced by ontology catalog, to provide automatic ontology visualization, automatic documentation, etc. Further, machines could automatically process your ontology. For instances, semantic search engines could automatically retrieve your ontology.
---	---

We encourage to follow the recommendation from this paper [11]. To reduce the learning curve, enclosed the code below.

```
1 <rdf:RDF
2   xmlns:dc="http://purl.org/dc/elements/1.1/"
3   xmlns:vann="http://purl.org/vocab/vann/"
4   xmlns:owl="http://www.w3.org/2002/07/owl#"
5   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
6   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
7   xmlns:dcterms="http://purl.org/dc/terms/"
8   xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
9 >
10 <owl:Ontology rdf:about="http://purl.org/iot/ontology/nameOnto#"
11   <owl:versionInfo>1.0</owl:versionInfo>
12   <vs:term_status>Work in progress</vs:term_status>
13   <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2018-02-16</dcterms:modified>
14   <dcterms:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2018-02-16</dcterms:issued>
15   <dc:title xml:lang="en">Name of your ontology Ontology</dc:title>
16   <dc:description rdf:resource="http://www.pathtexample/image.png"/>
17   <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
18     Find a nice name for your ontology
19   </rdfs:label>
20   <rdfs:comment xml:lang="en">
21     Description of your ontology is important for automatic, visualization documentation, etc.
22   </rdfs:comment>
23   <dc:creator xml:lang="en">XXX, Knoesis, USA</dc:creator>
24   <dc:contributor xml:lang="en">XXX, Knoesis, USA</dc:contributor>
25   <dc:description xml:lang="en">Describe the ontology</dc:description>
26   <vann:preferredNamespaceUri>http://purl.org/iot/ontology/nameOnto#</vann:preferredNamespaceUri>
27   <vann:preferredNamespacePrefix>Find a short prefix for your ontology</vann:preferredNamespacePrefix>
28   <dc:rights>Copyright - What is the license for your ontology</dc:rights>
29   <dcterms:bibliographicCitation xml:lang="en">
30     Citation of the paper describing your ontology
31   </dcterms:bibliographicCitation>
32 </owl:Ontology>
33 </rdf:RDF>
```



Figure 7. Adding Ontology metadata code within your ontology

Copy paste the following code to add ontology metadata. This code is for any ontology implemented with OWL/XML:

```

<rdf:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:vann="http://purl.org/vocab/vann/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
>

<owl:Ontology rdf:about="http://purl.org/iot/ontology/fiesta-iot#">
  <owl:versionInfo>1.0</owl:versionInfo>
  <vs:term status>Work in progress</vs:term status>
  <dcterms:modified
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2018-02-16</dcterms:modified>
  <dcterms:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2018-
02-16</dcterms:issued>
  <dc:title xml:lang="en">Name of your ontology Ontology</dc:title>
  <dc:description rdf:resource="http://www.pathexample/image.png"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Find a nice name for your ontology
  </rdfs:label>
  <rdfs:comment xml:lang="en">
    Description of your ontology is important for automatic,
visualization documentation, etc.
  </rdfs:comment>
  <dc:creator xml:lang="en">XXX, Knoesis, USA</dc:creator>
  <dc:contributor xml:lang="en">XXX, Knoesis, USA</dc:contributor>
  <dc:description xml:lang="en">Describe the ontology</dc:description>
  <vann:preferredNamespaceUri>http://purl.org/iot/ontology/nameOnto#</vann:pr
eferredNamespaceUri>
  <vann:preferredNamespacePrefix>sFind a short prefix for your
ontology</vann:preferredNamespacePrefix>
  <dc:rights>Copyright - What is the license for your ontology</dc:rights>
  <dcterms:bibliographicCitation xml:lang="en">
    Citation of the paper describing your ontology
  </dcterms:bibliographicCitation>
</owl:Ontology>

```



We recommend to follow the recommendation from this paper:

Pierre-Yves Vandenbussche and Bernard Vatant.
[Metadata recommendations for linked open data vocabularies](#) (2011).

4. Rule 5: Adding rdfs:label, rdfs:comment, dc:description for each concept and property (UPDATED March 2018)

```

▼<Declaration>
  <Class IRI="Availability"/>
</Declaration>
▼<Declaration>
  <Class IRI="AvailableBikes"/>
</Declaration>
▼<Declaration>
  <Class IRI="AvailableDesks"/>
</Declaration>

```



Figure 8. Bad practice example: There is a need to describe concepts or properties to know their meaning and definition

Recommendation example:

```

<Class rdf:about="http://www.example.com/ontologies/iot#Message">
  <rdf:comment>A Message is a instance of data that is sent by a source and that is received by a target</rdf:comment>
  <rdf:label>Message</rdf:label>
</Class>

```



Figure 9. Good Practice: Add rdf:label and rdf:comment to better describe concepts or properties

	Suggestion for improvement: Protégé tool needs to encourage ontology developers to create labels and comments. It is important to have such descriptions to have automatic ontology visualization, documentation, automatic alignment, etc.
---	---

5. [Rule 6: All classes should start with an Uppercase. The properties should start with a lowercase \(TO DO NEW February 2018\)](#)

6. [Rule 7: Submitting your ontology to ontology catalogs \(NEW February 2018\)](#)

To disseminate your ontology, we recommend to suggest your ontology to ontology catalogues. There are numerous ontology catalogues:

- [Linked Open Vocabularies \(LOV\)](#) designed by the Semantic Web community and requires that your ontology follows a set of best practices.
 - [Suggest you ontology on LOV here.](#)
- [Linked Open Vocabularies for Internet of Things \(LOV4IoT\)](#) references more than 400 ontology-base projects using sensors and classified according to IoT applicative domain (e. g.,

healthcare, smart city, etc.). The ontologies are also classified according to their best practices status (ontology shared online, ontology referenced by the LOV community, etc.)

- [Suggest your ontology on LOV4IoT here](#)
- **BioPortal**, with a focus on health and biomedical ontologies
- **OpenSensingCity** with a focus on smart city.
- **Ready4SmartCity**, seems not maintained anymore.

[Much more is explained here](#). With an interactive mindmap (see also figure below for a quick overview).

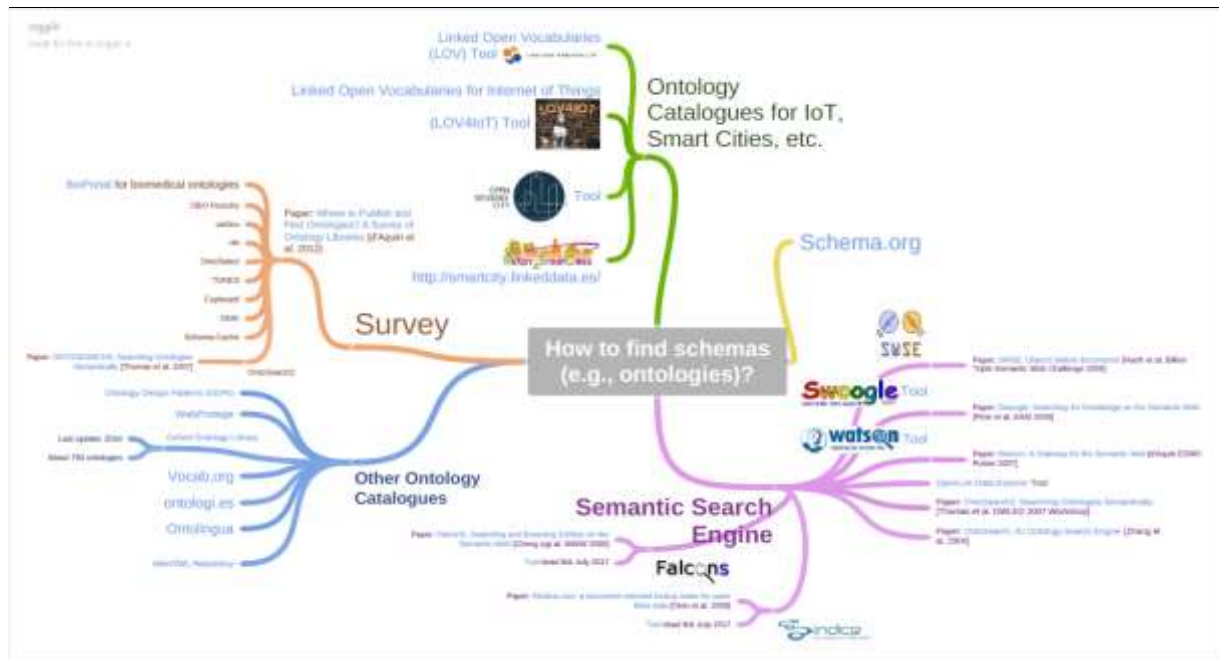


Figure 10. Ontology catalogue Mindmap with a set of tools to use.

7. Rule 8: Reusing and linking your ontology to other ontologies (TO DO - NEW February 2018)

Reuse an existing concepts and properties from an existing ontology/namespace (e.g., `ssn:Device`)

`owl:EquivalentClass`, `owl:sameAs`, `owl:equivalentProperty`,

TO DO: Explain the differences.

8. Rule 9: Deferenceable URI: copy paste the namespace URL of your ontology in a web browser to get the code

It means that when you copy paste the namespace URL of your ontology, you should get the code or the documentation.

Frequently we get the “404 – Not Found” Error:



Figure 11. Bad example: Copy paste the URL namespace of your ontology in a web browser (e.g., Chrome), you should not get the “404 – Not Found” Error, but get the ontology code or the documentation.

Good practice example:

`http://purl.org/iot/ontology/fiesta-iot#` redirects to `http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot.owl#` which provides the ontology code.

```

<rdf:RDF xmlns="http://purl.org/iot/ontology/fiesta-iot#" xmlns:iot-lite="http://purl.oclc.org/NET/UNIS/fiware/
xmlns:ns="http://creativecommons.org/ns#" xmlns:fiesta-iot="http://purl.org/iot/ontology/fiesta-iot#" xmlns:xsc
xmlns:mthreelite="http://purl.org/iot/vocab/m3-lite#" xmlns:ssn="http://purl.oclc.org/NET/ssnx/ssn#" xmlns:geo=
ns#" xmlns:onemtom="http://www.onem2m.org/ontology/Base_Ontology/base_ontology#" xmlns:qudt="http://data.qudt.c
xmlns:time="http://www.w3.org/2006/time#" xmlns:vann="http://purl.org/vocab/vann/" xmlns:dcterms="http://purl.c
xml:base="http://purl.org/iot/ontology/fiesta-iot">
  <owl:Ontology rdf:about="http://purl.org/iot/ontology/fiesta-iot#">
    <owl:versionInfo>3.1</owl:versionInfo>
    <dc:description rdf:resource="http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot.png"/>
    <dc:description rdf:resource="http://ontology.fiesta-iot.eu/ontologyDocs/resourcegraph.png"/>
    <dc:description rdf:resource="http://ontology.fiesta-iot.eu/ontologyDocs/observationgraph.png"/>
    <dc:description rdf:resource="http://ontology.fiesta-iot.eu/ontologyDocs/resourceannotation.png"/>
    <dc:description rdf:resource="http://ontology.fiesta-iot.eu/ontologyDocs/observationannotation.png"/>
    <dc:contributor xml:lang="en">Rachit Agarwal, Inria, Paris</dc:contributor>
    <dc:contributor xml:lang="en">David Gomez, UC, Santander</dc:contributor>
    <dc:contributor xml:lang="en">Tarek Elsaleh, UNIS, Surrey</dc:contributor>
    <dc:contributor xml:lang="en">Luis Sanchez, UC, Santander</dc:contributor>
    <dc:contributor xml:lang="en">Jorge Lanza, UC, Santander</dc:contributor>
    <dc:contributor xml:lang="en">Amelie Gyrard, NUIG, Galway</dc:contributor>
    <dc:contributor xml:lang="en">Garvita Bajaj, Inria, Paris (Past)</dc:contributor>
    <dc:description xml:lang="en">
      Many testbeds owning devices or applications interacting with the sensors, store their inherent observati
      independent and isolated from others that cannot directly interact with each other. Differences in the da
      order to ensure the compatibility. One has to understand different data formats and create the mapping be
      both while sending and receiving information. One method to accomplish interoperability is through the us
      Nevertheless, some of the relevant existing ontologies promise interoperability but: - do not address the
      reasoning, - many of them are domain-specific and cannot be applied across domains, - they have missing c
      ontologies do not follow best practices making it hard to correctly interpret and reuse concepts. In orde
      model) that follows best practices. FIESTA-IoT Ontology is a merge of concepts from various ontologies su
      divided into 2 parts: resource description and observations produced. One has to follow the Figure 2 and
      description and observation. Using such information testbeds can annotate their data to FIESTA-IoT specif
      we provide sample annotations as figure (see Figure 4 and Figure 5) for both resource graph and observati
      Lanza, L. Sanchez, N. Georgantas, V. Issarny, "Unified IoT Ontology to Enable Interoperability and Federa
      (https://hal.inria.fr/hal-01386917/document) - FIESTA-IoT Consortium (R. Agarwal editor), "Deliverable D3
      project FIESTA-IoT deliverable, 2016. - FIESTA-IoT Consortium (R. Agarwal editor), "Deliverable D3.1.1 Se
      FIESTA-IoT deliverable, 2016. We further acknowledge, testbed provider such as SmartSantander, UniS, KETI
      rachit.agarwal@inria.fr.
    </dc:description>
    <vann:preferredNamespaceUri>http://purl.org/iot/ontology/fiesta-iot#</vann:preferredNamespaceUri>
    <vann:preferredNamespacePrefix>fiesta-iot</vann:preferredNamespacePrefix>
    <dc:rights>Copyright EU H2020 FIESTA-IoT</dc:rights>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">FIESTA-IoT</rdfs:label>
    <dcterms:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2016-04-28</dcterms:issued>
  </rdf:comment xml:lang="en">

```



Figure 12. Good example: copy and paste an ontology namespace URL in a web browser to get the code

9. Rule 10 (TO DO)

10. Rule 11 (TO DO)

11. Rule 12 (TO DO)

12. Rule 13 (TO DO)

13. Rule 14 (TO DO)

14. Rule 15 (TO DO)

15. Rule 16: Registering your ontology on [prefix catalogs](#)

ⓘ Not secure | prefix.cc/dao

dao

<http://purl.org/dao#> ➔ ⁺¹₋₁

[Add alternative URI](#)

[ttl](#) [xml](#) [rdfa](#) [sparql](#) [txt](#) [json](#) [jsonld](#) [vann](#) | [lov](#) | [prefix.cc](#)

Not secure | yasgui.org

Query x +

yago

1

PREFIX

dbo:

<http://dbpedia.org/ontology/>

2

PREFIX

dbp:

<http://dbpedia.org/property/>

3

PREFIX

rdf:

<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

4

PREFIX

rdfs:

<http://www.w3.org/2000/01/rdf-schema#>

5

PREFIX

da

6

SELECT

da: <https://www.wowman.org/index.php?id=1&type=get#>

7

where {

dady: <http://purl.org/NET/dady#>

daia: <http://purl.org/ontology/daia/>

daiaserv: <http://purl.org/ontology/daia/Service/>

dailymed: <http://www4.wiwiss.fu-berlin.de/dailymed/resour

daisy: <http://www.daisy.org/z3998/2012/vocab/>

daml: <http://www.daml.org/2001/03/daml+oil#>

dannet: <http://www.wordnet.dk/owl/instance/2009/03/instar

daq: <http://purl.org/eis/vocab/daq#>

dash: <http://datashapes.org/dash#>

data: <http://data.odw.tw/>

datacite: <http://purl.org/spar/datacite/>

datacron: <http://www.datacron-project.eu/datAcron#>

datafaqs: <http://purl.org/twc/vocab/datafaqs#>

dataid: <http://dataid.dbpedia.org/ns/core#>

date: <http://contextus.net/ontology/ontomedia/misc/date#>

datex: <http://vocab.datex.org/terms#>

8

OP

9

OP

10

OP

11

OP

12

OP

13

FI

14

FI

Table

</>

This is probably because yasgui cached the prefixes locally in your browser. To test, you can open a private browser window (in chrome, that's 'ctrl+shift+n'), and try yasgui from there.

YASGUI

Not secure | yasgui.org

Query x +

http://dbpedia.org/sparql

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX dao:
4 SELECT dao: <http://purl.org/dao#>
5   ?sub ?pred ?obj .
6 }
7 LIMIT 10
```

YASGUI

Not secure | yasgui.org

WorkingMethodology teaching M3 knoesis ResearchProposal

Query x +

http://dbpedia.org/sparql

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX dao: <http://purl.org/dao#>
4 SELECT * WHERE {
5   ?sub ?pred ?obj .
6 }
7 LIMIT 10
```

V. How to check the design of your ontology (TO DO - NEW February 2018)

TO DO: How to check the conceptual level?

Otherwise check with the [Oops tool](#).

VI. Set of Tools to improve ontologies

1. Ontology Documentation with Parrot

We have encountered numerous errors when loading IoT ontologies.

- a) [Parrot Error: Unable to read input document: invalid mimeType "application/octet-stream" \(returned by URI\) for parrot](#)

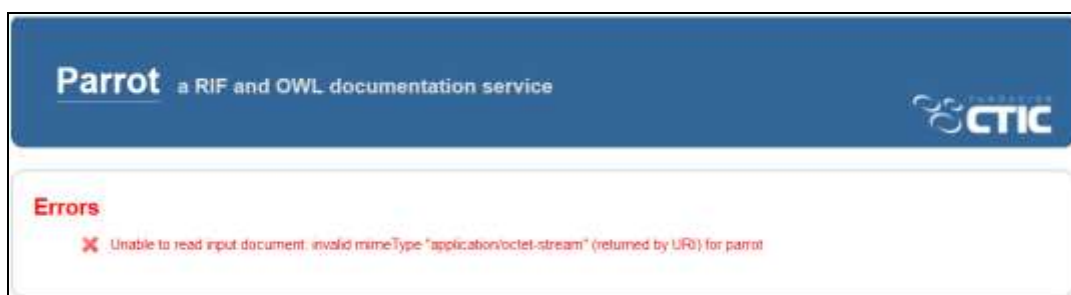


Figure 13. Unable to read input document: invalid mimeType "application/octet-stream" (returned by URI) for parrot

 <u>How to fix this?</u>	The mimeType returned for the Ontology URL is "application/octet-stream" (checked using curl -I), so Parrot is not able to identify (by content negotiation) the parser it use. In the web interface, you can select a concrete type (instead of "allow content negotiation"). After selecting "It is an OWL ontology", the documentation is generated.
--	---

- b) [Parrot Error: The ontology can be loaded but nothing is displayed](#)



Figure 14. No error encountered when loaded the ontology but nothing is displayed

	<u>How to fix this?</u>	The content-type returned by the server is "text/html", so it uses a parser for HTML. I guess that the file with the ontology URL is ending with .owl (even this file is served as "text/html"). When the parser is forced using the web interface ("It is an OWL ontology"), the documentation is generated.
--	---------------------------------------	---

c) [Parrot Error: I/O Error: Server returned HTTP response code: 403 for URL](#)

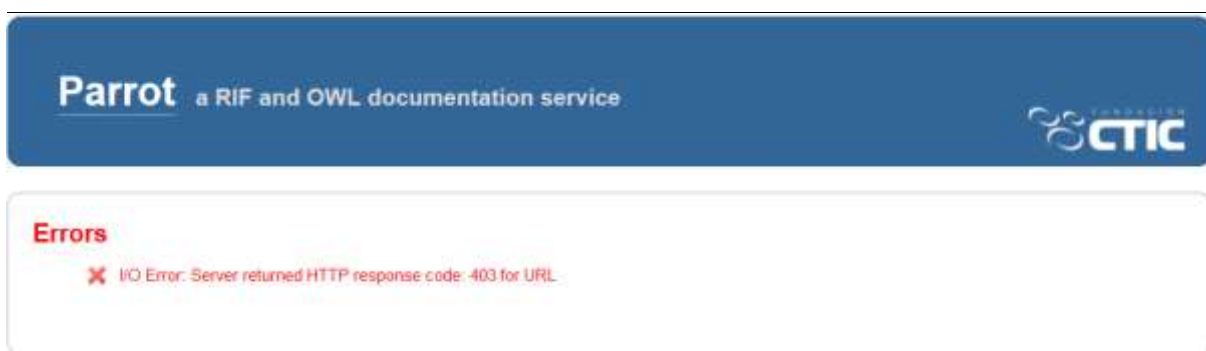


Figure 15. I/O Error: Server returned HTTP response code: 403 for URL

	<u>How to fix this?</u>	Using curl, this address returns a 200 OK code (with a strange content-type "application/rdf+xml"; see the '\' before the '+' symbol). After some testing, the server is not serving content if the "Accept" header does not include the "\+". I was able to generate the documentation in 2 steps. First, I have downloaded the file. Second, I have run Parrot using the "by file upload" tab and selecting directly the "It is an OWL ontology" parser.
--	---------------------------------------	---

- d) [Parrot Error: Unable to read input document: application/rdf+xml parse error: Content is not allowed in prolog.](#)



Figure 16. Unable to read input document: application/rdf+xml parse error: Content is not allowed in prolog.

 <u>How to fix this?</u>	The mimetype returned for the address ontology URL is "application/octet-stream" (checked using curl -I), so Parrot is not able to identify (by content negotiation) the parser to use. Also, the extension of the file is .owl and it looks like a Turtle file (.ttl). But in that case, Parrot is not returning anything if you change the parser.
---	--

2. Syntax Validator with TripleChecker

The RDF Triple-Checker tool helps find typos and common errors in RDF datasets or OWL ontologies.

a) TripleChecker Error: Possible match to "date"

6	property	http://purl.org/dc/elements/1.1/	description	OK	- Looks good.
6	property	http://purl.org/dc/elements/1.1/	creator	OK	- Looks good.
1	property	http://purl.org/dc/elements/1.1/	contributor	OK	- Looks good.
1	property	http://purl.org/dc/elements/1.1/	rights	OK	- Looks good.
1	property	http://purl.org/dc/elements/1.1/	issued	ERROR	- Possible match to "date" .. probable typo? (aff=5)
1	property	http://purl.org/dc/elements/1.1/	title	OK	- Looks good.
1	property	http://purl.org/dc/elements/1.1/	modified	ERROR	- Possible match to "date" .. probable typo? (aff=6)

Figure 17. Misuse of terms from Dublin Core namespace and date format with TripleChecker

 How to fix this?	You might have mixed between two namespaces: xmlns:dc=http://purl.org/dc/elements/1.1/ instead of xmlns:dcterms="http://purl.org/dc/terms/" "modified" an "issued" terms are from dcterms and not dc namespace. You might need to modify the code as well: <pre><dc:modified>2017-04-20</dc:modified> <dc:issued>2016-04-28</dc:issued></pre> by: <pre><dcterms:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2016- 04-28</dcterms:issued> <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2017- 04-20</dcterms:modified></pre>
Additional documentation	http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=elements#terms-issued http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=elements#terms-modified

b) TripleChecker Error: VERY close match to "license"

Count	Type	Namespace	Term	Looks Legit?
1	property	http://creativecommons.org/ns#	license	ERROR - VERY close match to "license" .. probable typo? (aff=1)

Figure 18. ERROR: VERY close match to "license"

 How to fix this?	TO DO
--	--------------

c) [TripleChecker Error - loading- No parser available](#)

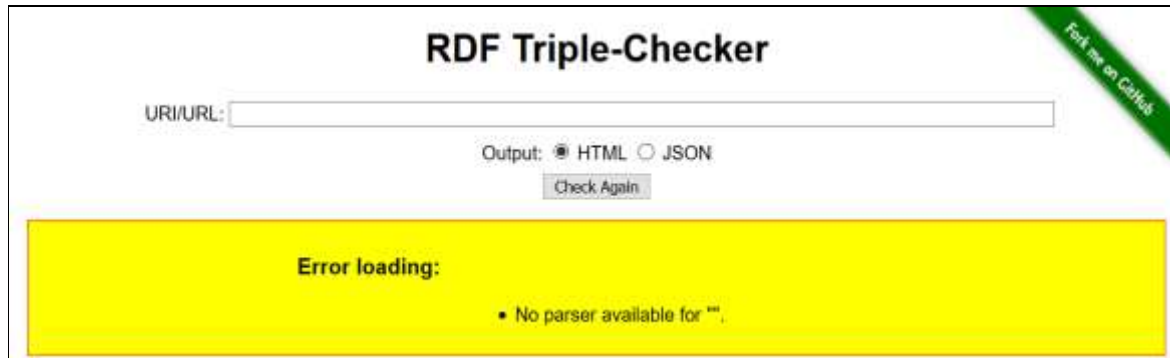


Figure 19. Error loading- No parser available

 <u>How to fix this?</u>	TO DO
---	--------------

3. [Ontology Visualization with WebVOWL](#)

We have encountered some errors when loading IoT ontologies.

a) [WebVOWL Error: There is nothing to visualize](#)

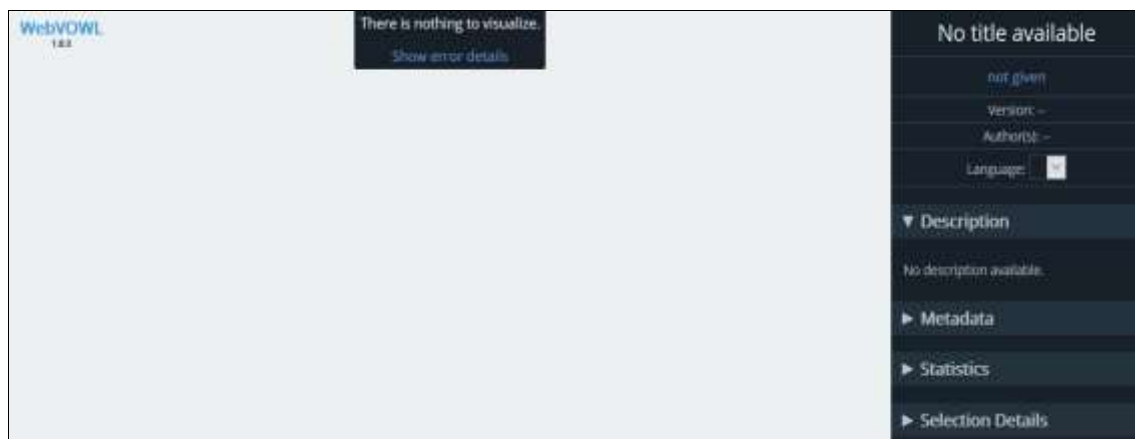


Figure 20. ERROR "There is nothing to visualize"

 <u>How to fix this?</u>	The ontology should passed the test with the OWL Manchester tool.
---	--

4. [Improving Ontology Design with OOPS](#)

Several kind of pitfalls can be encountered:

- Critical
- Important
- Minor

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P02: Creating synonyms as classes.	1 case Minor 🟡
Results for P04: Creating unconnected ontology elements.	16 cases Minor 🟡
Results for P07: Merging different concepts in the same class.	11 cases Minor 🟡
Results for P08: Missing annotations.	1013 cases Minor 🟡
Results for P11: Missing domain or range in properties.	27 cases Important ⚠️
Results for P12: Equivalent properties not explicitly declared.	4 cases Important ⚠️
Results for P13: Inverse relationships not explicitly declared.	54 cases Minor 🟡
Results for P20: Misusing ontology annotations.	1 case Minor 🟡
Results for P21: Using a miscellaneous class.	1 case Minor 🟡
Results for P22: Using different naming conventions in the ontology.	ontology* Minor 🟡
Results for P30: Equivalent classes not explicitly declared.	8 cases Important ⚠️
Results for P40: Namespace hijacking.	2 cases Critical 🚫
SUGGESTION: symmetric or transitive object properties.	12 cases

Figure 21. OOPS Report with Critical pitfalls

 How to fix this?	Please check the documentation on OOPS web site to fix the errors.
--	---

a) [OOPS Error: Pitfall 36 - URI contains file extension](#)

Results for P36: URI contains file extension.	ontology* Minor 🟡
This pitfall occurs if file extensions such as ".owl", ".rdf", ".ttl", ".n3" and ".rdxml" are included in an ontology URI. This pitfall is related with the recommendations provided in [9]. *This pitfall applies to the ontology in general instead of specific elements.	

Figure 22. OOPS Pitfall 36 URI contains file extension

 How to fix this?	The URL tested has something like: http://www.onto.org/ontology/Base_Ontology/BaseOntology.owl Better to remove the .owl extension Be careful, when you copy paste the URL of the ontology on the web browser, we still need to get the ontology file and not the 404 page not found.
--	---

b) [OOPS Error: Pitfall 37 - Ontology not available on the Web](#)

Results for P37: Ontology not available on the Web.

ontology* | Critical

This pitfall occurs when the ontology code (OWL encoding) or its documentation (HTML document) is missing when looking up its URI. This pitfall deals with the first point from the Linked Data star system that states "On the web" ([10] and [11]). Guidelines in [12] also recommends to "Publish your vocabulary on the Web at a stable URI". This pitfall is also related to the problems listed in [8] and [5].

*This pitfall applies to the ontology in general instead of specific elements.

Figure 23. OOPS Error: P37: Ontology not available on the Web

 How to fix this?	TO DO
---	-------

1. Ontology Matching with LogMap web service (NEW: April 2019)

LogMap error encountered:

We have detected an error when parsing the input ontologies. Possible causes: the given URI is not accessible, the URI contains a non permitted character, or the given ontology format is not currently accepted.

 How to fix this?	In this case, 2 URLs tested: • http://purl.org/kao# which redirects to: • https://raw.githubusercontent.com/gyrard/ontologies/master/kaoV2.owl# Probably because, of the HTTPS, because if we load the file itself, the matching will work.
---	--

2. Ontology Documentation with LODE

We have encountered numerous errors when loading IoT ontologies.

- a) LODE Error - Reason: An empty sequence is not allowed as the value of variable \$rdf

LODE error Reason: An empty sequence is not allowed as the value of variable \$rdf

Figure 24. Reason: An empty sequence is not allowed as the value of variable \$rdf

 <u>How to fix this?</u>	TO DO
--	-------

- b) LODE Error - Reason: A sequence of more than one item is not allowed as the @select attribute of xsl:sort

LODE error Reason: A sequence of more than one item is not allowed as the @select attribute of xsl:sort
--

Figure 25. LODE Error - Reason: A sequence of more than one item is not allowed as the @select attribute of xsl:sort

 <u>How to fix this?</u>	TO DO
--	-------

- c) [LODE Error - Reason: org.xml.sax.SAXParseException; lineNumber: 1; columnNumber: 1; Content is not allowed in prolog.](#)

LODE error

Reason: org.xml.sax.SAXParseException; lineNumber: 1; columnNumber: 1; Content is not allowed in prolog.

Figure 26. Reason: org.xml.sax.SAXParseException; lineNumber: 1; columnNumber: 1; Content is not allowed in prolog.

 <u>How to fix this?</u>	TO DO
--	-------

- d) [LODE Error - The source can't be downloaded in any permitted format. # Connection reset # Connection reset # Connection reset # Connection reset # Connection reset # Connection reset](#)

LODE error

Reason: The source can't be downloaded in any permitted format. # Connection reset # Connection reset # Connection reset # Connection reset # Connection reset # Connection reset

Figure 27. LODE Error - The source can't be downloaded in any permitted format. # Connection reset # Connection reset # Connection reset # Connection reset # Connection reset # Connection reset

 <u>How to fix this?</u>	TO DO
--	-------

- e) [LODE Error - The source can't be downloaded in any permitted format. # Received fatal alert: protocol_version](#)

LODE error

Reason: The source can't be downloaded in any permitted format. # Received fatal alert: protocol_version

Figure 28. LODE error. The source can't be downloaded in any permitted format. # Received fatal alert: protocol_version

 <u>How to fix this?</u>	TO DO
--	-------

VII. Suggesting Ontologies on LOV

An error occurred:

WARN [line: 7, col: 9] Bad IRI: <http://example.com/QoSQoI.ow/#> Code: 57/REQUIRED_COMPONENT_MISSING in HOST: A component that is required by the schema is missing.

Figure 29. LOV Suggest Error – Suggesting an ontology to LOV

Bad IRI: <http://example.com> Code: 57/REQUIRED_COMPONENT_MISSING in HOST: A component that is required by the schema is missing.

	<u>How to fix this?</u>	TO DO
---	--------------------------------	-------

VIII. Frequently Seen Bad Practices

- The default namespace has not been changed (frequently with Protege). A protégé extension would be nice to encourage to have a good namespace.
- Ontologies only available within a PDF file! Argggggg!!!!
- See more on this document:
<http://sensormeasurement.appspot.com/publication/OneM2MBestPractices.pdf>

IX. Useful Links

By the creator of TripleChecker: “Linked Data Basics for Techies”:

http://web.archive.org/web/20160308191327/http://openorg.ecs.soton.ac.uk/wiki/Linked_Data_Basics_for_Techies

X. References

[1] Vandenbussche, P.Y., Ateazing, G.A., Poveda-Villalón, M., Vatan, B.: Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web. Semantic Web Journal (2015)

[2] McBride, B.: Jena: A semantic web toolkit. Internet Computing, IEEE (2002)

[3] 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 (IJSWIS) (2014)

[4] 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)

- [5] Lohmann, S., Link, V., Marbach, E., Negru, S.: WebVOWL: Web-based visualization of ontologies. In: Knowledge Engineering and Knowledge Management. Springer (2014)
- [6] Berrueta, D., Fernández, S., Frade, I.: Cooking http content negotiation with vapour. In: Proceedings of 4th Workshop on Scripting for the Semantic Web (SFSW2008), Citeseer (2008)
- [7] Gyrard, A., Serrano, M., Atemezing, G.: Semantic Web Methodologies, Best Practices and Ontology Engineering Applied to Internet of Things. In: WF-IOT 2015, World Forum on Internet of Things, Milan, Italy, IEEE (2015)
- [8] 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, European Research Cluster on the Internet of Things, AC4 (2015)
- [9] 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)
- [10] Amelie Gyrard, Christian Bonnet, 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)
- [11] Pierre-Yves Vandenbussche and Bernard Vatant. Metadata recommendations for linked open data vocabularies (2011)
- [12] Murdock et al. [Semantic Interoperability for the Web of Things](#) White paper (2016)