INPUT CONTRIBUTION

Group Name:*	MAS WG
Title:*	Semantic Web best practices. Semantic Web Guidelines for domain knowledge interoperability to build the Semantic Web of Things
Source:*	Eurecom, Amelie Gyrard, Christian Bonnet
Contact:	Amelie Gyrard, gyrard@eurecom.fr, Christian Bonnet, bonnet@eurecom.fr
Date:*	2014-04-07
Abstract:*	This contribution proposes to describe the semantic web best practices, semantic web tools, and existing domain ontologies for uses cases (smart home and health).
Agenda Item:*	Tbd
Work item(s):	MAS
Document(s) Impacted*	Study of Existing Abstraction & Semantic Capability Enablement Technologies for consideration by oneM2M.
Intended purpose of document:*	☐ Decision ☐ Discussion ☒ Information ☐ Other <specify>
Decision requested or recommendation:*	This is an informative paper proposed by the French Eurecom institute as a guideline to MAS contributors on Semantic web best practices, as it was suggested during MAS#9.3 call. Amélie Gyrard is a new member in oneM2M (via ETSI PT1).

1. Content

2. List of Figures

2. Scope

The present document describes semantic web guidelines such as the best practices, interoperability issues, the semantic tools, and domain ontologies already existing to build the Semantic Web of Things (SWoT), a new field to combine Semantic Web technologies and Internet of Things.

We aim to bridge the gap between the Semantic Web and Internet of Things communities.

3. References

[1] Kenza Abdelaziz, Philippe Roose, Sebastien Laborie, France Anglet, Bernd Amann, Tayeb Lemlouma, and France Lannion. Toward a context-aware and automatic evaluation of elderly dependency in smart homes and cities.

[2] Robert Barta, Christina Feilmayr, Birgit Pröll, Christoph Grün, and Hannes Werthner. Covering the semantic space of tourism: an approach based on modularized ontologies. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, page 1. ACM, 2009.

[3] Diego Berrueta and Jon Phipps. Best practice recipes for publishing rdf vocabularies-w3c working group note. *World Wide Web Consortium (WC), Aug.,.: http://www. w3. org/TR/2008/NOTE-swbp-vocab-pub-20080828/.(Cit. on pp.,)*, 2008.

[4] D. Bonino and F. Corno. Dogont-ontology modeling for intelligent domotic environments. *The Semantic Web-ISWC 2008*, pages 790–803, 2008.

[5] Paul Brandt, Twan Basten, Sander Stuiik, Vinh Bui, Paul de Clercq, Lus Ferreira Pires, and Marten van Sinderen. Semantic interoperability in sensor applications making sense of sensor data. Ín *Computational Intelligence in Healthcare and e-health (CICARE), 2013 IEEE Symposium on*, pages 34–41. IEEE, 2013.

[6] F Calore, D Lombardi, Enrico Mussi, Pierluigi Plebani, and Barbara Pernici. Retrieving substitute services using semantic annotations: A foodshop case study. In *Business Process Management Workshops*, pages 508–513. Springer, 2008.

[7] Harry Chen. An intelligent broker architecture for context-aware systems. *PhD proposal in Computer Science, University of Maryland, Baltimore, USA*, 2003.
[8] Harry Chen, Tim Finin, and Amupam Joshi. Semantic web in the context broker architecture. Technical report, DTIC Document, 2005.
[9] Harry Chen, Tim Finin, and Anupam Joshi. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(03):197–207, 2003.
[10] Harry Chen, Tim Finin, and Anupam Joshi. The soupa ontology for pervasive computing. In *Ontologies for agents: Theory and experiences*, pages 233–258. Springer, 2005.
[11] Harry Chen, Tim Finin, Anupam Joshi, Lalana Kagal, Filip Perich, and Dipanjan Chakraborty. Intelligent agents meet the semantic web in smart spaces. *Internet Computing, IEEE*, 8(6):69–79, 2004.
[12] Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi. Soupa: Standard ontology for ubiquitous and pervasive applications. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 258–267. IEEE, 2004.
[13] Hung-Yu Chien, Shyr-Kuen Chen, Ching-Yang Lin, Jia-Ling Yan, Wei-Chen Liao, Huan-Yi Chu, Kuan-Ju Chen, Bo-Fan Lai, and Yi-Ting Chen. Design and implementation of zigbee-ontology-based exhibit guidance and recommendation system.
[14] Luke Ruth David Wood, Marsha Zaidman and Michael Hausenblas. *Linked Data. Structured Data on the Web.* Manning Publications, 2014.
[15] Janna Hastings, Werner Ceusters, Barry Smith, and Kevin Mulligan. The emotion ontology: enabling interdisciplinary research in the affective sciences. In *Modeling and Using Context*, pages 119–123. Springer, 2011.
[16] Tom Heath and Christian Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.
[17] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 345–354. ACM, 2013.
[18] Mark Hennessy, Chris Oentojo, and Steven Ray. A framework and ontology for mobile sensor platforms in home health management. In *Engineering of Mobile-Enabled Systems (MOBS), 2013 1st International Workshop on the*, pages 31–35. IEEE, 2013.
[19] Alan Jovic, Dragan Gamberger, and Goran Krstacic. Heart failure ontology. *Bio-algorithms and med-systems*, 7, 2011.
[20] M.J. Kofler, C. Reinisch, and W. Kastner. An intelligent knowledge representation of smart home energy parameters. In *Proceedings of the World Renewable Energy Congress (WREC 2011), Linköping, Sweden*, 2011.
[21] Fatiha Latfi, Bernard Lefebvre, and Céline Descheneaux. Ontology-based management of the telehealth smart home, dedicated to elderly in loss of cognitive autonomy. In *OWLED*, volume 258, 2007.
[22] Freddy Lécué, Simone Tallevi-Diotallevi, Jer Hayes, Robert Tucker, Veli Bicer, Marco Sbodio, and Pierpaolo Tommasi. Star-city: Semantic traffic analytics and reasoning for city?
[23] HANG LIU. Improving semantic interoperability in remote patient monitoring applications. 2013.
[24] Juan López, Rosa Gil, Roberto Garcá, Idoia Cearreta, and Nestor Garay. Towards an ontology for describing emotions. *Emerging Technologies and Information Systems for the Knowledge Society*, pages 96–104, 2008.
[25] Laurent Mazuel, Jean Charlet, et al. Alignement entre des ontologies de domaine et la snomed: trois études de cas. In *Actes de la 20ème conférence Ingénierie des Connaissances-IC2009*, 2009.
[26] Georgios Meditskos, Stamatia Dasiopoulou, Vasiliki Efstathiou, and Ioannis Kompatsiaris. Ontology patterns for complex activity modelling. In *Theory, Practice, and Applications of Rules on the Web*, pages 144–157. Springer, 2013.
[27] Philippe Morignot and Fawzi Nashashibi. An ontology-based approach to relax traffic regulation for autonomous vehicle assistance. *CoRR*, abs/1212.0768, 2012.
[28] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[29] Federica Paganelli and Dino Giuli. An ontology-based context model for home health monitoring and alerting in chronic patient care networks. In *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, volume 2, pages 838–845. IEEE, 2007.
[30] Federica Paganelli and Dino Giuli. An ontology-based system for context-aware and configurable services to support home-based continuous care. *Information Technology in Biomedicine, IEEE Transactions on*, 15(2):324–333, 2011.
[31] Evangeline Pollard, Philippe Morignot, and Fawzi Nashashibi. An Ontology-based Model to Determine the Automation Level of an Automated Vehicle for Co-Driving. In *16th International Conference on Information Fusion*, Istanbul, Turquie, July 2013.
[32] Andreas Pomarolli, Stefan Anderlik, and Josef Küng. R2rif-rule integration plugin for protégé owl. In *Computer Aided Systems Theory–EUROCAST 2011*, pages 137–144. Springer, 2012.
[33] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, and Koen De Bosschere. Towards an extensible context ontology for ambient intelligence. In *Ambient intelligence*, pages 148–159. Springer, 2004.
[34] Alan Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. Owl pizzas: Practical experience of teaching owl-dl: Common errors & common patterns. In *Engineering Knowledge in the Age of the Semantic Web*, pages 63–81. Springer, 2004.
[35] C. Reinisch, M.J. Kofler, F. Iglesias, and W. Kastner. Thinkhome energy efficiency in future smart homes. *EURASIP Journal on Embedded Systems*, 2011:1, 2011.
[36] Daniele Riboni and Claudio Bettini. Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.
[37] Daniele Riboni and Claudio Bettini. Owl 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing*, 7(3):379–395, 2011.
[38] Daniele Riboni, Linda Pareschi, Laura Radaelli, and Claudio Bettini. Is ontology-based activity recognition really effective? In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 427–431. IEEE, 2011.
[39] Michele Ruta, Floriano Scioscia, Filippo Gramegna, and Eugenio Di Sciascio. A mobile knowledge-based system for on-board diagnostics and car driving assistance. In *UBICOMM 2010, The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 91–96, 2010.
[40] Oumy Seye, Catherine Faron-Zucker, Olivier Corby, and Corentin Follenfant. Bridging the gap between rif and sparql: Implementation of a rif dialect with a sparql rule engine. *AImWD 2012*, page 19, 2012.
[41] Paul Staroch. A weather ontology for predictive control in smart homes. Master's thesis, 2013.
[42] Thanos G Stavropoulos, Dimitris Vrakas, Danai Vlachava, and Nick Bassiliades. Bonsai: a smart building ontology for ambient intelligence. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, page 30. ACM, 2012.
[43] Chuan-Jun Su, Chang-Yu Chiang, and Meng-Chun Chih. Ontological knowledge engine and health screening data enabled ubiquitous personalized physical fitness (ufit). *Sensors*, 14(3):4560–4584, 2014.
[44] Mará Taboada, Mará Jesús Sobrido, Verónica Colombo, and Belén Pilo. A semantic web approach to integrate phenotype descriptions and clinical data. In *Knowledge Representation for Health-Care*, pages 16–26. Springer, 2011.
[45] Pierre-Yves Vandenbussche and Bernard Vatant. Metadata recommendations for linked open data vocabularies. *Version*, 1:2011–12, 2011.
[46] Johann Vincent, Christine Porquet, Maroua Borsali, and Harold Leboulanger. Privacy protection for smartphones: an ontology-based firewall. In *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication*, pages 371–380. Springer, 2011.
[47] Johann Vincent, Christine Porquet, Idriss Oulmakhzoune, et al. Ontology-based privacy protection for smartphone: A firewall implementation. In *International Conference on Secure Networking and Applications (ICSNA)*, 2011.
[48] Zachary Wemlinger and Lawrence Holder. The cose ontology: Bringing the semantic web to smart environments. In *Toward Useful Services for Elderly and People with Disabilities*, pages 205–209. Springer, 2011.

[49]Konlakorn Wongpatikaseree, Mitsuru Ikeda, Marut Buranarach, Thepchai Supnithi, Azman Osman Lim, and Yasuo Tan. Location-based concept in activity log ontology for activity recognition in smart home domain. In *Semantic Technology*, pages 326–331. Springer, 2013.

[50]Wen Yao, C Chu, Akhil Kumar, and Zang Li. Using ontology to support context awareness in healthcare. In *Proceedings of the 19th Workshop on Information Technologies and Systems: Dec 14-15 2009; Phoenix, AZ, USA*, 2009.

[51] Wen Yao and Akhil Kumar. Conflexflow: Integrating flexible clinical pathways into clinical decision support systems using context and rules. *Decision Support Systems*, 55(2):499–515, 2013.

[52] Jun Zhao et al. Publishing chinese medicine knowledge as linked data on the web. *Chinese medicine*, 5(1):1–12, 2010.

4. Definitions, symbols, abbreviations and acronyms

4.1 Definitions

SPIN: A W3C recommendation to design semantic-based rules.

SPARQL: A query language for RDF.

4.2 Symbols


Good practices are explained


You can encounter some difficulties or errors by using tools.

4.3 Acronyms

LOV	Linked Open Vocabularies
SWoT	Semantic Web of Things
SPIN	(SPARQL Inferencing Notation)
SPARQL	SPARQL Protocol and RDF Query Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
OWL	Ontology Web Language
SWRL	Semantic Web Rule Language
RIF	Rule Interchange Format

5. Introduction

Semantic Web of Things (SWoT) is a new field to combine Semantic Web technologies and Internet of Things. Firstly, domain experts constantly redefined new domain knowledge (ontology and rules) without considering the existing ones. Secondly, domain experts are not aware of the semantic web best practices or semantic web tools. The OneM2M standard is relevant to spread the semantic web best practices and encourage domain experts to choose semantic web tools to develop the domain knowledge, in order to reuse easily they ontology-based works. Further, there is a need to standardize domain ontologies.

The following guidelines should be taken into account when defining new domain knowledge.

6. Semantic web guidelines

6.1 Design your ontology

You can find tutorials to design your first ontology:

- Ontology Development 101: A Guide to Creating Your First Ontology [28]
- OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors and Common Patterns [34]

6.2 Describe domain knowledge at least written in English

Describe your domain knowledge at least in **English**. You can describe labels and comments in various languages if needed. In the Figure 1, as you can see, if you are not familiar with the Chinese, Spanish or German language you cannot reuse these works.

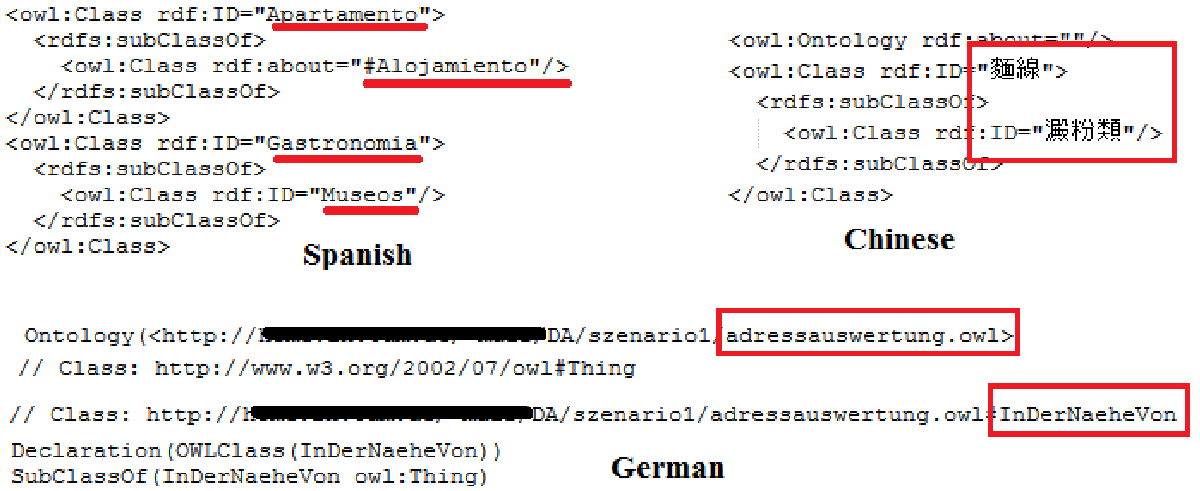


Figure 1 Ontology only written in Chinese, Spanish or German is not easily reusable

The **good practice** is to describe your ontology at least in English and if needed in another language as depicted in Figure 2. Document the domain knowledge (concepts, properties, instances) with human-friendly labels and comments (rdfs:label and rdfs:comment, dcterms:description) is recommended.


```
<owl:Class rdf:ID="Brushing_teeth">
  <owl:disjointWith>
    <owl:Class rdf:ID="Brushing_hair"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Brushing_clothes"/>
  </owl:disjointWith>
  <rdfs:label xml:lang="en">Brushing teeth</rdfs:label>
  <rdfs:label xml:lang="fr">Se brosser les dents</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Brushing"/>
  </rdfs:subClassOf>
</owl:Class>
```

Figure 2 Lafti et al. [21] design an health ontology both in English and French

6.3 Ontology best practices

6.3.1 Choose a good namespace

As you can see in the Figure 3, the ontology does not have a good name since it is called unnamed.owl

sisinflab[redacted]/bag/iDriveSafe.txt

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.doom-srl.com/unnamed.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about="http://www.doom-srl.com/unnamed.owl"/>
  <owl:Class rdf:about="http://www.doom-srl.com/unnamed.owl#ESP">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://www.doom-srl.com/unnamed.owl#Safety_Device"/>
    </rdfs:subClassOf>
  </owl:Class>
```

Figure 3 The ontology does not have a good namespace

The **good practice** is to have the same URI for both the namespace and the ontology location as depicted in the Figure 7. This mechanism is called **URI deferencable**. For example, the URI http://www.gdst.uqam.ca/Documents/Ontologies/HIT/Task_SH_Ontology.owl entered on a web browser gives access to the ontology.

6.3.2 Publish online the ontology

Publish online the ontology on your server. Choose a cool URI[1].

The OWL file is directly accessible through the Web not in a zip file or other as depicted in the Figure 4.

[1] http://www.w3.org/Provider/Style/URI.html

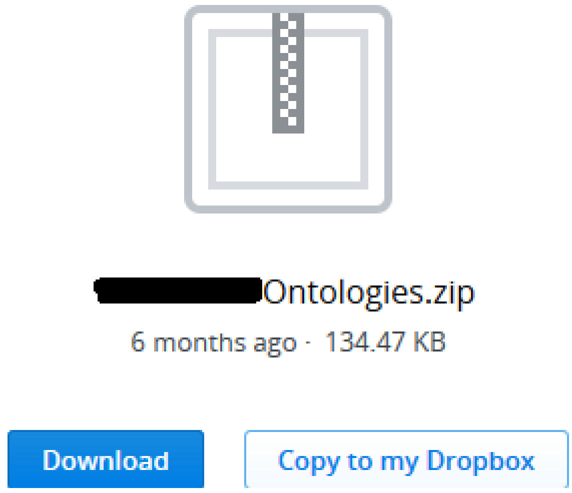


Figure 4 Bad practice since ontologies are in a zip file

6.3.3 Ontology URI deferencable, Content Negociation Problem

Once the ontology is published online, the ontology can be submitted to the LOV project.

Frequently, domain experts encountered the problem **Content Negociation Problem** as depicted in the Figure 5.

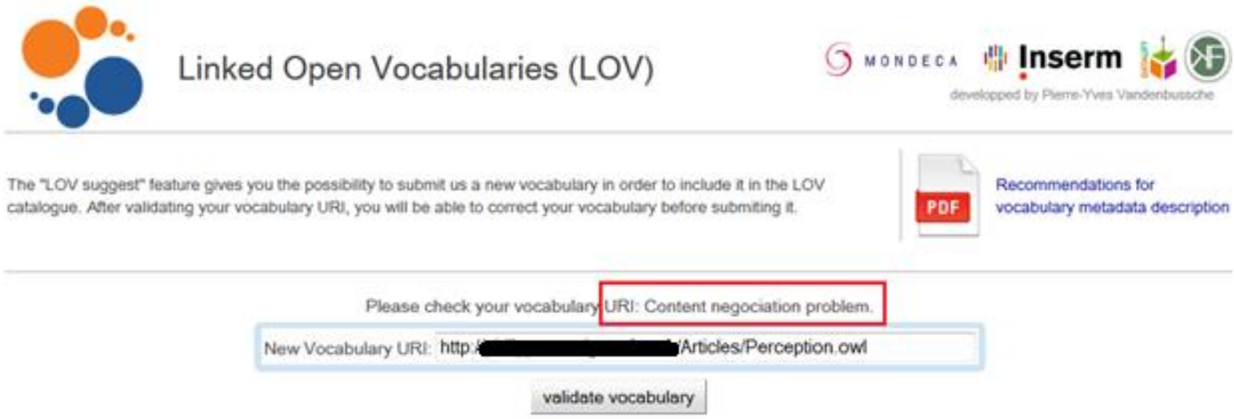
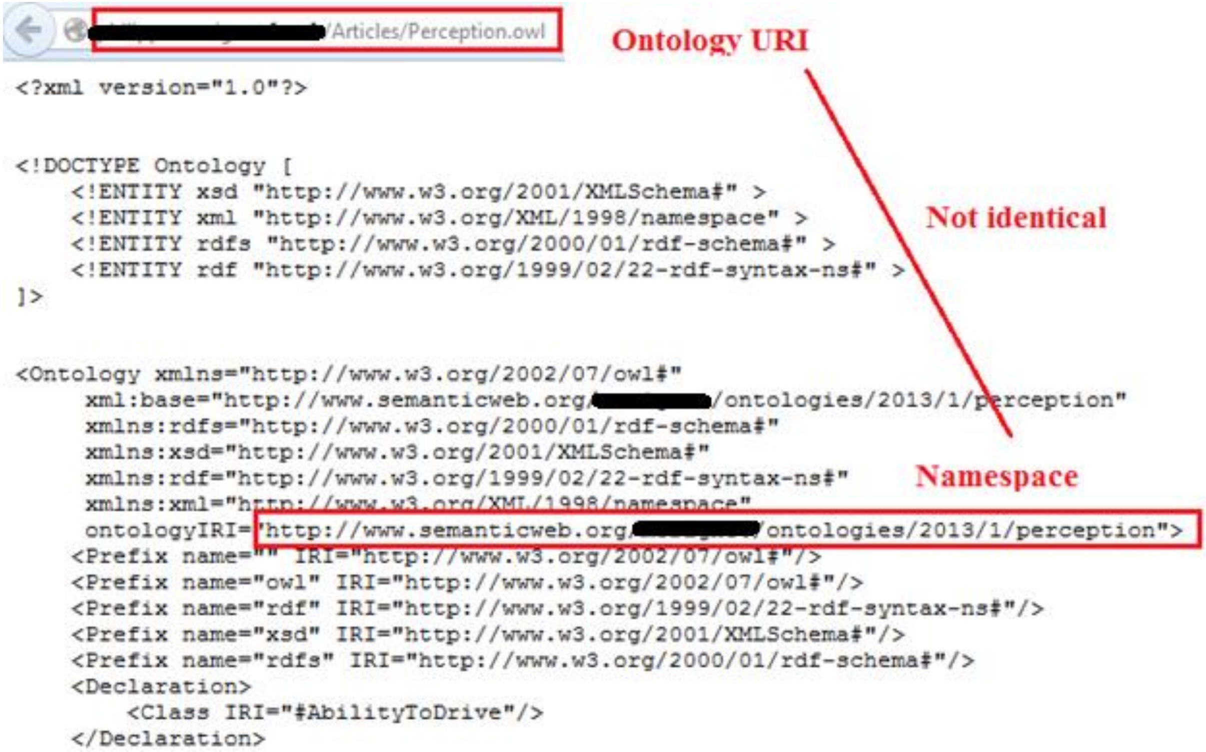


Figure 5 Content negociation problem

When we look up the **namespace** of the ontology on a Web browser, we should find the ontology. The namespace of the ontology should be the same that the location of the ontology, it is called **URI deferencable**. In the Figure 6, this is not the case the namespace URI and the ontology URI are not identical, this is why the LOV project generated the **context negociation** error.



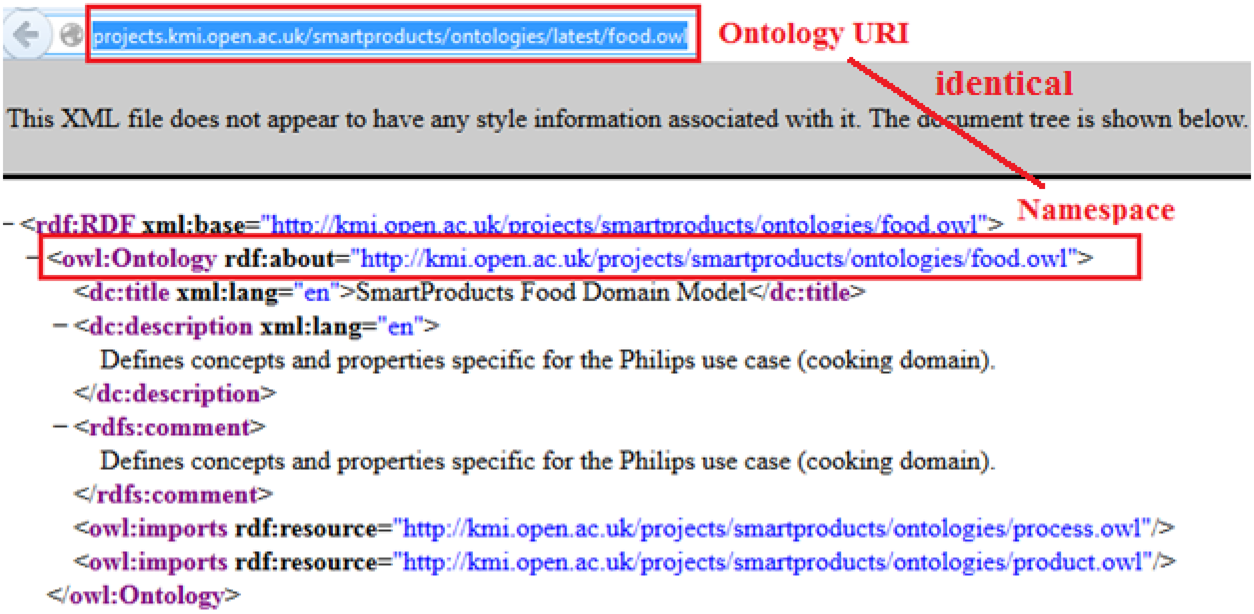

```
<?xml version="1.0"?>


<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>


<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.semanticweb.org[redacted]/ontologies/2013/1/perception"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     ontologyIRI="http://www.semanticweb.org[redacted]ontologies/2013/1/perception">
    <Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Declaration>
        <Class IRI="#AbilityToDrive"/>
    </Declaration>
```

Figure 6 The namespace and the ontology URI are not identical

The **good practice** is to have the same URI for both the namespace and the ontology location as depicted in the Figure 7. This mechanism is called **URI deferencable**.




```
This XML file does not appear to have any style information associated with it. The document tree is shown below.

- <rdf:RDF xml:base="http://kmi.open.ac.uk/projects/smartproducts/ontologies/food.owl">
  - <owl:Ontology rdf:about="http://kmi.open.ac.uk/projects/smartproducts/ontologies/food.owl">
      <dc:title xml:lang="en">SmartProducts Food Domain Model</dc:title>
    - <dc:description xml:lang="en">
        Defines concepts and properties specific for the Philips use case (cooking domain).
      </dc:description>
    - <rdfs:comment>
        Defines concepts and properties specific for the Philips use case (cooking domain).
      </rdfs:comment>
      <owl:imports rdf:resource="http://kmi.open.ac.uk/projects/smartproducts/ontologies/process.owl"/>
      <owl:imports rdf:resource="http://kmi.open.ac.uk/projects/smartproducts/ontologies/product.owl"/>
    </owl:Ontology>
```

Figure 7 The namespace and the ontology URI are identical

6.3.4 Reuse with existing ontologies

Reuse domain knowledge rather than reinventing them:

- The ontology should reuse existing ontologies wherever possible.
- Add owl:equivalentClass for common concepts already defined in existing ontologies
- The class or properties are those from the ontologies referenced on LOV.
- Link common concept (owl:equivalentClass or rdfs:subClassOf) with well-known ontologies (e.g., Person is already described in FOAF)
- You can always extend an ontology to fit your needs

```
<owl:Class rdf:about="http://paul.staroch.name/thesis/SmartHomeWeather.owl#Heat">
    <owl:equivalentClass>
        <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="http://paul.staroch.name/thesis/SmartHomeWeather.owl#WeatherPhenomenon"/>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="http://paul.staroch.name/thesis/SmartHomeWeather.owl#hasTemperatureValue"/>
                    <owl:someValuesFrom>
                        <owl:Class>
                            <owl:intersectionOf rdf:parseType="Collection">
                                <owl:Restriction>
                                    <owl:onProperty rdf:resource="&muo;measuredIn"/>
                                    <owl:hasValue rdf:resource="&ucum;unit/temperature/degree-Celsius"/>
                                </owl:Restriction>
                                <owl:Restriction>
                                    <owl:onProperty rdf:resource="&muo;numericalValue"/>
                                    <owl:someValuesFrom>
                                        <rdfs:Datatype>
                                            <owl:onDatatype rdf:resource="&xsd;float"/>
                                            <owl:withRestrictions rdf:parseType="Collection">
                                                <rdf:Description>
                                                    <xsd:minExclusive rdf:datatype="&xsd;float">30.0</xsd:minExclusive>
                                                </rdf:Description>
                                            </owl:withRestrictions>
                                        </rdfs:Datatype>
                                    </owl:someValuesFrom>
                                </owl:Restriction>
                            </owl:intersectionOf>
                        </owl:Class>
                    </owl:someValuesFrom>
                </owl:Restriction>
            </owl:intersectionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:comment rdf:datatype="&xsd;string">A WeatherPhenomenon describing a temperature of more than 30 degrees Celsius.
    <rdfs:comment rdf:resource="http://paul.staroch.name/thesis/SmartHomeWeather.owl#Temperature"/>
    <rdfs:comment rdf:resource="http://paul.staroch.name/thesis/SmartHomeWeather.owl#WeatherPhenomenon"/>
    <rdfs:comment rdf:resource="http://paul.staroch.name/thesis/SmartHomeWeather.owl#hasTemperature"/>
</owl:Class>
```

Good practice: use exising unit ontologies (MUO, UCUM)

Figure 8 Staroch et al. define a smart home ontology related to the weather [41].

Some ontologies are not longer maintained but cannot be ignored.

This is the case for SWEET implemented by the NASA which design about 6000 concepts in 200 separate ontologies.

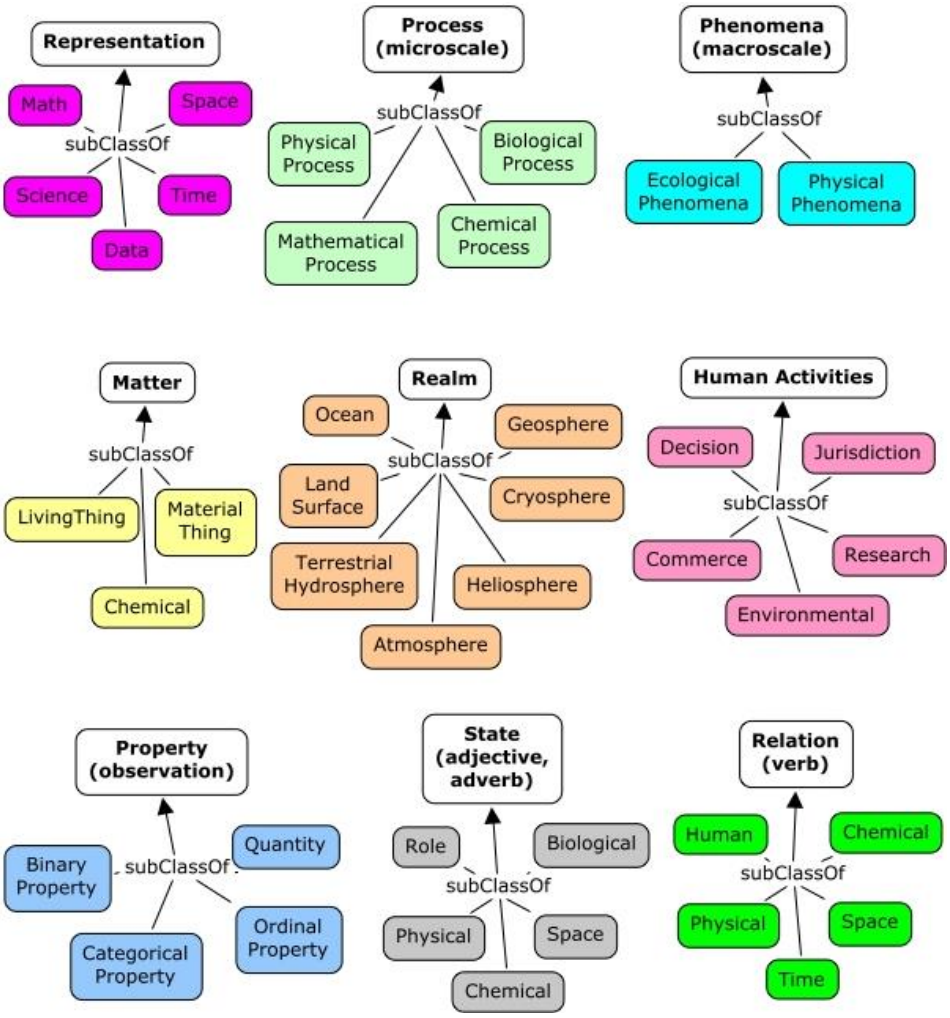


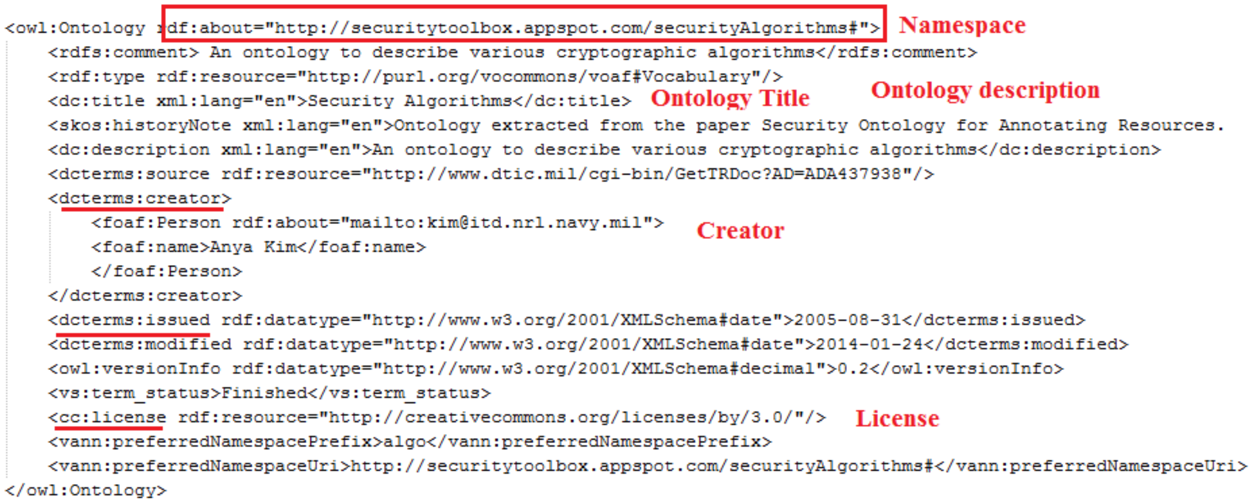
Figure 9 SWEET ontologies

Some ontologies are still maintained but is linked to ontologies which are not maintained anymore, for example the emotion ontology [15] which is based on the OBO ontology.

6.3.5 Ontology metadata: LOV recommendation

Reference your ontology on LOV(see section Ontology catalogue)

- Add ontology metadata recommended by LOV as depicted in the Figure 10 [45]
- Metadata Recommendations For Linked Open Data Vocabularies
- A code example is available (See Annexe A: Ontology LOV metadata)




```
<owl:Ontology rdf:about="http://securitytoolbox.appspot.com/securityAlgorithms#">
    <rdfs:comment> An ontology to describe various cryptographic algorithms</rdfs:comment>
    <rdf:type rdf:resource="http://purl.org/vocommons/voaf#Vocabulary"/>
    <dc:title xml:lang="en">Security Algorithms</dc:title>
    <skos:historyNote xml:lang="en">Ontology extracted from the paper Security Ontology for Annotating Resources.
    <dc:description xml:lang="en">An ontology to describe various cryptographic algorithms</dc:description>
    <dcterms:source rdf:resource="http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA437938"/>
    <dcterms:creator>
        <foaf:Person rdf:about="mailto:kim@itd.nrl.navy.mil">
        <foaf:name>Anya Kim</foaf:name>
        </foaf:Person>
    </dcterms:creator>
    <dcterms:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2005-08-31</dcterms:issued>
    <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2014-01-24</dcterms:modified>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.2</owl:versionInfo>
    <vs:term_status>Finished</vs:term_status>
    <cc:license rdf:resource="http://creativecommons.org/licenses/by/3.0/"/>
    <vann:preferredNamespacePrefix>algo</vann:preferredNamespacePrefix>
    <vann:preferredNamespaceUri>http://securitytoolbox.appspot.com/securityAlgorithms#</vann:preferredNamespaceUri>
</owl:Ontology>
```

Figure 10 Ontology metadata recommended by LOV

Frequently domain experts encountered some errors when submitting their ontology to LOV.

If this is the case, check:

- Test the ontology URL on [Vapour]
- Test the ontology URL on [RDF Triple-Checker]
- The ontology best practices

6.3.6 Server-side configuration, Vapour

Vapour is a link data validator to check whether the data are correctly published according to the semantic web guidelines, as defined by the Linked Data principles, the Best Practice Recipes and the Cool URIs.

Vapour checks three tasks:

- 1st request while dereferencing resource URI without specifying the desired content type (HTTP response code should be 200)
- 1st request while dereferencing resource URI without specifying the desired content type (HTTP response code should be 200)
- 1st request while dereferencing resource URI without specifying the desired content type (Content type should be 'application/rdf+xml')

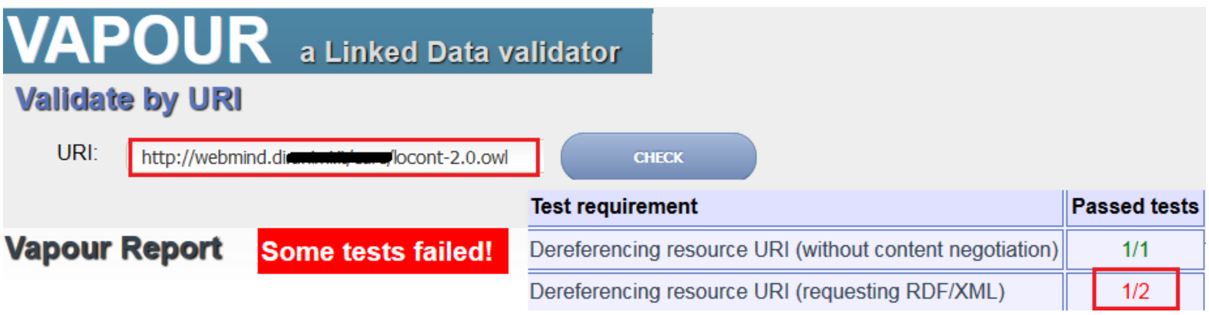


Figure 11 Vapour tool failed

The domain experts have to correct the error "1st request while dereferencing resource URI without specifying the desired content type (Content type should be 'application/rdf+xml'): Failed".

The solution is to configure the server. For instance for Apache server you can change the httpd.conf configuration file and add the following line.

```
AddType application:rdf+xml .rdf
```

Or you can add this information in the .htaccess file in the directory on the server where the RDF files are placed.

This is a main issue to achieve this task, since some authors share their ontologies:

- On a personal web page, they cannot control the server
- Use google app engine

Some domain experts try to host their ontologies on GitHub, it was a good idea, but it generates an error on Vapour:

```
IlegalLocationValue: the value of the location header in the response
(https://github.com/ngankam/ontology/blob/master/instrusion_description_in_
wsn) is not an absolute URI (see the RFC 2616, section 14.30)
```

https://tools.ietf.org/html/rfc2616

```
14.30 Location

   The Location response-header field is used to redirect the recipient
   to a location other than the Request-URI for completion of the
   request or identification of a new resource. For 201 (Created)
   responses, the Location is that of the new resource which was created
   by the request. For 3xx responses, the location SHOULD indicate the
   server's preferred URI for automatic redirection to the resource. The
   field value consists of a single absolute URI.

       Location       = "Location" ":" absoluteURI

   An example is:

       Location: http://www.w3.org/pub/WWW/People.html

      Note: The Content-Location header field (section 14.14) differs
      from Location in that the Content-Location identifies the original
      location of the entity enclosed in the request. It is therefore
      possible for a response to contain header fields for both Location
      and Content-Location. Also see section 13.10 for cache
      requirements of some methods.
```



Vapour was unable to complete the request due to the following exception: **see section 14.30 in RFC2616**

```
IlegalLocationValue: the value of the location header in the response (https://github.com/[redacted]/ontology/blob/master/instrusion_description_in_wsn) is not an absolute URI
```

Complete traceback (see/hide):

```
  File "/opt/instancias/vapour/vapour/apps/vapour/../vapour/cup/webclient.py", line 86, in GET
    recipes.checkRecipes(store, resourceToCheck, validatorOptions)
  File "/opt/instancias/vapour/vapour/apps/vapour/../vapour/teapot/recipes.py", line 32, in checkRecipes
    checkWithoutAcceptHeader(graph, resource, validatorOptions)
  File "/opt/instancias/vapour/vapour/apps/vapour/../vapour/teapot/recipes.py", line 48, in checkWithoutAcceptHeader
    runScenario(graph, resource, scenarioDescription, requestedContentType, validatorOptions, "GET")
  File "/opt/instancias/vapour/vapour/apps/vapour/../vapour/teapot/recipes.py", line 81, in runScenario
    (rootTestSubject, httpResponse) = followRedirects(graph, "dereferencing " + resource['description'], resource['uri'], requestedContentType, httpMethod, validatorOptions)
  File "/opt/instancias/vapour/vapour/apps/vapour/../vapour/teapot/httpdialog.py", line 35, in followRedirects
    raise IlegalLocationValue(url)
```


Figure 12 Vapour error indicates to see the RFC 2616, section 14.30

6.3.7 Provide an ontology documentation

- Parrot is a web service, there is nothing to install. Less than 30 minutes to add a documentation to your dataset or ontology.
- Neologism. Need to install the software
- SpecGen . Need to install the software


BEVON: Beverage Ontology

This Version

http://rdfs.co/bevon/0.7 [HTML] [RDF/XML] [Turtle]

Latest Version

http://rdfs.co/bevon/

Previous Version

http://rdfs.co/bevon/0.6

This vocabulary is *under development*.

Copyright © 2013-2014 James G. Kim Some Rights Reserved.

This work is licensed under a Creative Commons License.

Table of Contents

- Introduction
- Changes From Previous Version
- Namespace
- Terms Grouped by Theme
- Summary of Terms
- Vocabulary Classes
- Vocabulary Properties
- Examples
- License

Figure 13 Documentation example

6.3.8 Validate with OOPS

The Oops tool will detect common errors. An example is to avoid to have two ideas in a same concept as depicted in the Figure 14.

```
<owl:Class rdf:ID="Eating_or_drinking">
    <rdfs:label>Eating_or_drinking</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Kitchen_Activity" />
</owl:Class>
```

Figure 14 Do not describe 2 ideas in the same concept

6.3.9 Validate your ontology with semantic web validators

They are more and more tools implemented by the semantic web community to detect common errors when developing your RDF data or ontologies.

- RDF Validator is used to check your RDF documents as depicted in the Figure 15.
- OWL Validator is used to check your OWL documents.
- OOPS! (OntOlogy Pitfall Scanner!) is a tool to detect common ontology errors as depicted in the Figure 16.
- The RDF Triple-Checker tool helps find typos and common errors in RDF data
- Vapour is a link data validator to check whether the data are correctly published according to the semantic web guidelines, as defined by the Linked Data principles, the Best Practice Recipes and the Cool URIs.
- RDFAbout is a RDF Validator and Converter between the RDF/XML format and N3 (Notation 3 or N-Triples Turtle).

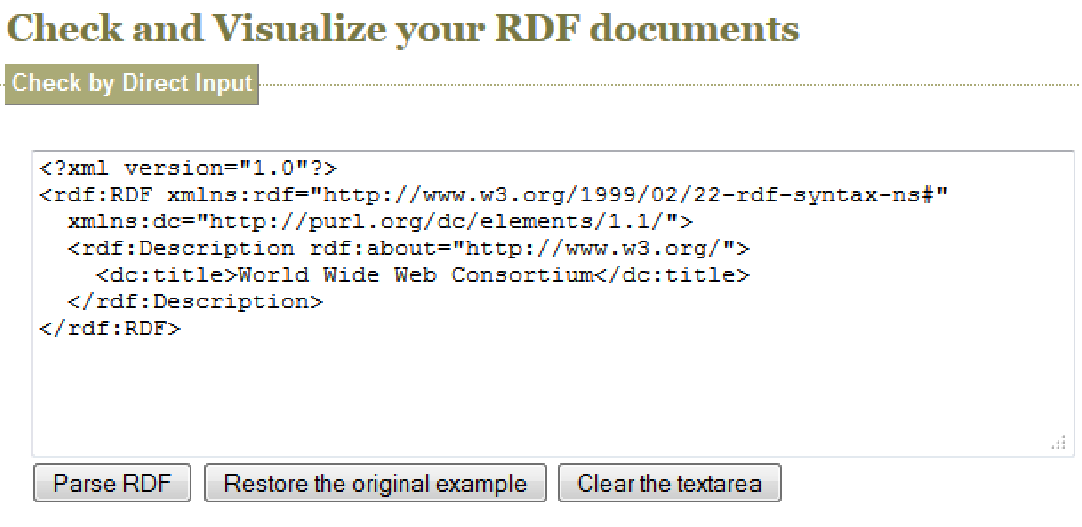


Figure 15 RDF validator

Evaluation results

This results have been generated from DBpedia Ontology 3.8 on 3rd January of 2014. These results might be outdated if the original ontology changes.

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 P04: Creating unconnected ontology elements.	16 cases | Minor
Results for P07: Merging different concepts in the same class.	3 cases | Minor

A class is created whose identifier is referring to two or more different concepts. An example of this type of pitfall is to create the class "StyleAndPeriod", or "ProductOrService".

• This pitfall appears in the following elements:
› http://schema.org/LandmarksOrHistoricalBuildings
› http://schema.org/CollegeOrUniversity
› http://schema.org/StadiumOrArena

Results for P08: Missing annotations.	2366 cases | Minor
Results for P11: Missing domain or range in properties.	528 cases | Important
Results for P12: Missing equivalent properties.	74 cases | Important
Results for P13: Missing inverse relationships.	1003 cases | Minor
Results for P20: Misusing ontology annotations.	4 cases | Minor
Results for P22: Using different naming criteria in the ontology.	ontology* | Minor
Results for P27: Defining wrong equivalent relationships.	1 case | Critical
Results for P30: Missing equivalent classes.	16 cases | Important
Results for P31: Defining wrong equivalent classes.	25 cases | Critical
Results for P32: Several classes with the same label.	9 cases | Minor
Results for P34: Untyped class.	51 cases | Important
Results for P35: Untyped property.	30 cases | Important
SUGGESTION: symmetric or transitive object properties.	131 cases

Figure 16 The Oops tool detects errors when developing ontologies

6.4 Dataset best practices

Some documents to create a well-designed dataset:

- Linked Data: Evolving the Web into a Global Data Space. [16]. This book introduces the principles for publishing Linked Data or designed Linked Data applications. 2011
- Linked Data. Structured Data on the Web. 2014 [14]
- Best Practice Recipes for Publishing RDF Vocabularies. (More difficult to read) [3]

- How to Publish Linked Data on the Web
- Linked Data (design issues)
- Linked Open Data

Some tools to publish your data:

- D2R server enables to publish your database schema as a SPARQL endpoint.
- Jena fuseki
- SPARQL endpoint
- Reference your dataset on DataHub and other related tools (see section Dataset catalogue).

Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods.

Figure 17 Linked Open Data Best practices

Publishing descriptions of a data set:

- Semantic SiteMap to add metadata to the dataset (e.g., sparql endpoint)
- void (Vocabulary of Interlinked Datasets) is a standard vocabulary for describing datasets

To digitally sign your data you can use the NG4J, a Named Graphs API for Jena.

7. Ontology interoperability

We referenced in this section usual tools to design ontologies used by domain experts.

- Protégé is the most used ontology free editor tool to design a new ontology as depicted in the Figure 41 and proposes various plugin for ontology visualization, writing rules, etc.
- OWL API
- TopBraid
- More tools are referenced in the section Ontology editors, semantic API or framework.

7.1 Protégé

Protégé is a popular tool for ontology editing and representation.

```
<owl:Class rdf:ID="Attraction">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:ID="SouvenirShop"/>
        <owl:Class rdf:about="#Attraction"/>
        <owl:Class rdf:ID="BikeRental"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
```

Figure 18 Chien et al. [13] design a tourism ontology with Protege

7.2 OWL API

OWL API as depicted in the **Error! Reference source not found.**

```
<SubClassOf>
    <Class IRI="#Sensor"/>
    <Class IRI="#Indicating_meas_instrument"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#True_quant_value"/>
    <Class IRI="#Quantity_value"/>
</SubClassOf>
<DisjointClasses>
    <Class IRI="#Detector"/>
    <Class IRI="#Sensor"/>
```

Figure 19 Ontology designed with OWL API [5]

7.3 TopBraid

TopBraid is a commercial solution to build semantic web and linked data applications

```
<owl:Class rdf:ID="Facial">
  <rdfs:subClassOf rdf:resource="#Behavioural"/>
</owl:Class>
<owl:Class rdf:ID="BiologicalSensor">
  <rdfs:subClassOf rdf:resource="http://www.loa-
cnr.it/ontologies/DUL.owl#BiologicalObject"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Sensor"/>
  </rdfs:subClassOf>
</owl:Class>
```

Figure 20 Lopez et al. designed an emotion ontology [24] with TopBraid


```
<owl:Ontology rdf:ID="tourexp">
  <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Created with TopBraid Composer</owl:versionInfo>
  <owl:imports rdf:resource="http://topbraid.org/wgs84_pos"/>
</owl:Ontology>
<owl:Class rdf:ID="Entorno">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Imagen">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="AlojamientoRural">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Alojamiento"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Archivos">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Patrimonio"/>
  </rdfs:subClassOf>
</owl:Class>
```

Figure 21 Bujan et al. designed a tourism ontology with TopBraid and the rdf/xml syntax [2] and not in english

```
hsl:ECGReading
      rdf:type owl:Class ;
      rdfs:subClassOf hsl:MedicalReading .

hsl:GlucoseReading
      rdf:type owl:Class ;
      rdfs:subClassOf hsl:MedicalReading ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:cardinality "1"^^xsd:nonNegativeInteger ;
                owl:onProperty hsl:bloodGlucoseLevel
              ] ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:cardinality "1"^^xsd:nonNegativeInteger ;
                owl:onProperty hsl:units
              ] .
```

Figure 22 Hennessy, Ray et al. designed an emotion ontology with TopBraid and the turtle syntax [18]

8. Rules interoperability

There is a need to work on the interoperability of the different implementation of ontologies and rules generated by software and semantic tools.

Various languages have been referenced to describe the semantic web rules:

- SWRL (Semantic Web Rule Language) is frequently used by domain experts since it is easy to use and already implemented by software. **This language is not advocated by the semantic web community.**
- SPIN (SPARQL Inferencing Notation) is **advocated by semantic web experts since it is a W3C recommendation since 2013.**
- RIF (Rule Interchange Format). Usual software used by domain experts do not implement RIF.
- Rules describes as restriction in the ontologies

8.1 OWL rules interoperability

Frequently rules are directly described as restrictions in ontologies. Interoperability issues have been discovered for interlining these rules: the syntax is not identic according to the software used, they do not use the exact same term (snowy, snow, snowy weather state).

Example how to combine rules related to the same concept snow:

Rule 1 (smart home domain): Snowy = belowOrZeroTemperature and Precipitation [20] [35] (Figure 25) is implemented with the OWL API.

Rule 2 (smart city domain): Snowy = belowOrZeroTemperature and Precipitation [22] (Figure 24) is implemented with the OWL API.

Rule 3 (transport domain): Snow -> safety device ABS, ESP, and snow chains [39] (Figure 27) is implemented with OWLed2[2]

8.1.1 OWL restrictions

```
<owl:Class rdf:ID="Eating_or_drinking">
    <rdfs:label>Eating_or_drinking</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Kitchen_Activity" />
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:cardinality>
            <owl:onProperty rdf:resource="#has_hasFood" />
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#has_hasFood" />
            <owl:allValuesFrom rdf:resource="#Food" />
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:cardinality>
            <owl:onProperty rdf:resource="#has_hasPosture" />
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#has_hasPosture" />
            <owl:allValuesFrom rdf:resource="#Sit" />
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 23 Wongpatikaseree et al. [49] defines rules to infer activities

[2] http://www.doom-srl.it/index.php?option=com_content&task=view&id=20&Itemid=30&lang=en

8.1.2 OWL rules with OWL API

```
<EquivalentClasses>
    <Class IRI="#SnowyWeatherState"/>
    <ObjectIntersectionOf>
        <Class IRI="#WeatherState"/>
        <ObjectIntersectionOf>
            <ObjectComplementOf>
                <ObjectSomeValuesFrom>
                    <ObjectProperty IRI="http://www.ibm.com/SCTC/ontology/CoreSpatioTemporalDataSensorOntology.owl#hasPhenomenon"/>
                    <Class IRI="#NoPrecipitation"/>
                </ObjectSomeValuesFrom>
            </ObjectComplementOf>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="http://www.ibm.com/SCTC/ontology/CoreSpatioTemporalDataSensorOntology.owl#hasPhenomenon"/>
                <Class IRI="#BelowOrZeroTemperature"/>
            </ObjectSomeValuesFrom>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="http://www.ibm.com/SCTC/ontology/CoreSpatioTemporalDataSensorOntology.owl#hasPhenomenon"/>
                <Class IRI="#Precipitation"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </ObjectIntersectionOf>
</EquivalentClasses>
```

Figure 24 The Star-city ontology [22] defines rules to infer if it is snowy


```
<!-- https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/WeatherOntology.owl#SnowyWeatherState -->

<owl:Class rdf:about="&WeatherOntology;SnowyWeatherState">
    <owl:equivalentClass>
        <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="&WeatherOntology;WeatherState"/>
                <owl:Class>
                    <owl:intersectionOf rdf:parseType="Collection">
                        <owl:Class>
                            <owl:complementOf>
                                <owl:Restriction>
                                    <owl:onProperty rdf:resource="&WeatherOntology;hasWeatherPhenomenon"/>
                                    <owl:someValuesFrom rdf:resource="&WeatherOntology;NoPrecipitation"/>
                                </owl:Restriction>
                            </owl:complementOf>
                        </owl:Class>
                        <owl:Restriction>
                            <owl:onProperty rdf:resource="&WeatherOntology;hasWeatherPhenomenon"/>
                            <owl:someValuesFrom rdf:resource="&WeatherOntology;BelowOrZeroTemperature"/>
                        </owl:Restriction>
                        <owl:Restriction>
                            <owl:onProperty rdf:resource="&WeatherOntology;hasWeatherPhenomenon"/>
                            <owl:someValuesFrom rdf:resource="&WeatherOntology;Precipitation"/>
                        </owl:Restriction>
                    </owl:intersectionOf>
                </owl:Class>
            </owl:intersectionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&WeatherOntology;hasWeatherCondition"/>
            <owl:hasValue rdf:resource="&WeatherOntology;Snow"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 25 The ThinkHome ontology [20] [35] defines rules to infer if it is snowy in the building automation domain.

8.1.3 OWL rules with Protégé

```
<owl:Class rdf:ID="FreshPoultry">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#isPerishable"/>
      </owl:onProperty>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >true</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Poultry"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="FrozenPoultry"/>
  </owl:disjointWith>
</owl:Class>
```

Figure 26 Food tiscaly ontology[6] implemented with Protege

8.1.4 OWL rules with OWLed2

```
<owl:Class rdf:about="http://www.doom-srl.com/[redacted]#Snow">
  <rdfs:subClassOf rdf:resource="http://www.doom-srl.com/[redacted]#Weather"/>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="http://www.doom-srl.com/[redacted]#Weather"/>
        <owl:Restriction>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="http://www.doom-srl.com/[redacted]#hasSafety_Device"/>
          </owl:onProperty>
          <owl:allValuesFrom>
            <owl:Class>
              <owl:intersectionOf rdf:parseType="Collection">
                <owl:Class rdf:about="http://www.doom-srl.com/[redacted]#Snow_Chains"/>
                <owl:Class rdf:about="http://www.doom-srl.com/[redacted]#ABS"/>
                <owl:Class rdf:about="http://www.doom-srl.com/[redacted]#ESP"/>
              </owl:intersectionOf>
            </owl:Class>
          </owl:allValuesFrom>
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="http://www.doom-srl.com/[redacted]#hasSensor_Speed"/>
          </owl:onProperty>
          <owl:allValuesFrom rdf:resource="http://www.doom-srl.com/[redacted]#Low_Speed"/>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

Figure 27 Ruta et al. [39] describe safety devices (abs, esp, and snow chains) related to the snow

8.1.5 OWL rules with topBraid

```
hsl:PulseReading
      rdf:type owl:Class ;
      rdfs:subClassOf hsl:MedicalReading ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:cardinality "1"^^xsd:nonNegativeInteger ;
                owl:onProperty hsl:pulseRate
              ] ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:cardinality "1"^^xsd:nonNegativeInteger ;
                owl:onProperty hsl:pulseUnits
              ] ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:cardinality "1"^^xsd:nonNegativeInteger ;
                owl:onProperty hsl:spO2
              ] ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:cardinality "1"^^xsd:nonNegativeInteger ;
                owl:onProperty hsl:spO2Units
              ] .
```

Figure 28 Hennessy et al. [18] designed an health ontology in turtle with TopBraid

8.2 SWRL (Semantic Web Rule Language)

SWRL (Semantic Web Rule Language), based on OWL and RuleML, is the most popular rule language since it is easy to use and used by domain experts. **This language is not advocated by the semantic web community.**

Unfortunately, the syntax varying according to the software or inference engine employed (OWL restrictions in the ontology, Jena, SWRL Tab protege, Pellet, Fact++, etc.):

- JenaRules, JenaRules wiki
- SWRL Tab (Plugin Protege) [O'Connor 2006]
- SWRL DL Safe Rule that restricts rules to operate on only known individuals of ontology.
- SWRLJess Tab (Plugin Protege)
- SWRL-IQ (Plugin Protege)
- SQWRL (Plugin Protege)
- SWRLDroolsTab (Plugin Protege)

8.2.1 Jena rules

```
#Forbid any action made by a service provider with an invalid
certificate.
[policy1: (?s rdf:type id:ServiceProvider) , (?a rdf:type id:Action),
(?c rdf:type id:InvalidCertificate), (?s id:hasCertificate ?c)
-> (id:policy1 id:forbids ?a)]
```

Figure 29 Vincent et al. [46] [47] design Jena rules in the security domain

8.2.2 SWRL and DLSafeRule

SWRL DL Safe Rule restricts rules to operate on only known individuals of ontology.

These SWRL rules are developed with the SWOOPS tool. The syntax is again different.

```
<DLSafeRule>
    <Body>
        <ClassAtom>
            <Class IRI="#Foggy"/>
            <Variable IRI="urn:swrl#c"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#FullyManual"/>
            <Variable IRI="urn:swrl#a"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#LongitudinalHighPrecision"/>
            <Variable IRI="urn:swrl#b"/>
        </ClassAtom>
    </Body>
    <Head>
        <ClassAtom>
            <Class IRI="#DynamicSetSpeedType"/>
            <Variable IRI="urn:swrl#a"/>
        </ClassAtom>
    </Head>
</DLSafeRule>
```

Figure 30 Morignot et al. [31] [27] design DLSafeRule in the transportation system

8.3 SPIN (SPARQL Inferencing Notation)

SPIN (SPARQL Inferencing Notation) is advocated by semantic web experts since it is a W3C recommendation**:**

- Jena SPIN rules (Jena ARQ API)
- SPIN SPARQL syntax
- SPARQL CONSTUCT (equivalent to SWRL rules)
- SPINMap[3] is used by Hennessy et al. [18] in a health-based work.
- SPARQL Motion

SPARQL query	SELECT ?user ?Goal WHERE { ?User hasBodyFat ?I. }
SPIN rule 1	CONSTRUCT {?user hasHighBodyFat ?I.} WHERE{ ?this hasBodyFat ?I. FILTER (?I > 24). }
SPIN rule 2	CONSTRUCT { ?I value_of_MedicalSign ?M. } WHERE { ?user hasHighBodyFat ?I. }
SPIN rule 3	CONSTRUCT { ?user hasGoal ?G. } WHERE { ?G hasMedicalSign ?M. ?I value_of_MedicalSign ?M. ?user hasHighBodyFat ?I. }
SPIN rule 4	CONSTRUCT { ?user hasPreferredExercise ?E. } WHERE { ?user hasGoal ?G. ?G rdfs:label “reduce-body-fat”. ?E hasTypeAerobics “Aerobic”. FILTER (?user hasAbilitySwimming “false”. ?E hasTypeWatersports “false”). }

Figure 31 Su et al. [43] design SPIN rules in the health domain

[3] http://composing-the-semantic-web.blogspot.fr/2011/04/spinmap-sparql-based-ontology-mapping.html

```
CONSTRUCT {
  ?y a BedExit; //SpecialisationType
    isSpecialisedBy ?x.
}
WHERE{
  ?x a NightSleep; //ContextType
    hasStartTime ?st1;
    hasEndTime ?et1;
    hasActor ?p.
  ?y a OutOfBed; //SpecialisedType
    hasStartTime ?st2;
    hasEndTime ?et2;
    hasActor ?p.
  FILTER(:contains(?st1, ?et1, ?st2, ?et2))
}
```

```
CONSTRUCT {
  ?new a Nocturia; //CompositeType
    hasStartTime ?st1;
    hasEndTime ?et2;
    hasActor ?p;
    hasSubActivity ?x;
    hasSubActivity ?y.
}
WHERE{
  ?x a BedExit; //SubActivityType
    hasStartTime ?st1;
    hasEndTime ?et1;
    hasActor ?p.
  ?y a InBathroom; //SubActivityType
    hasStartTime ?st2;
    hasEndTime ?et2;
    hasActor ?p.
  FILTER(:contains(?st1, ?et1, ?st2, ?et2))
  BIND(:newURI(?x, ?y) as ?new)
}
```

Figure 32 Efstathiou et al. [26] deisn SPIN rules in the smart home domain.

8.4 RIF (Rule Interchange Format)

RIF (Rule Interchange Format). Usual software used by domain experts do not implement RIF:

- RIF2SPARQL and RIF validator [40]
- Paper: R2RIF - Rule Integration Plugin for Protege OWL [32] - No plugin found
- RIF implementations

9. Domain ontologies interesting for the OneM2M uses cases

We referenced domain ontologies which could be reused and extended with new concepts for the use cases. The following ontologies are available and authors are improving the ontologies according to the semantic web guidelines. To find the corresponding ontology URL or more ontologies, you can search on this web page: http://www.sensormeasurement.appspot.com/?p=ontologies and the LOV project (http://lov.okfn.org/dataset/lov/).

9.1 Building Automation Ontologies

Bonino et al. [4] design the DogOnt ontology[4], referenced by LOV, is one of the first ontology respecting the semantic web guidelines in the building automation domain. They describe the following concepts:

- Building environment (`Room` in a house such as `Bathroom, Bedroom, DiningRoom, Kitchen, LivingRoom, Lobby, StorageRoom`)
- Building thing: controllable (`fridge, oven, coffee maker, alarm clock, printer`) or not (`wall, floor`).
- Functionality (`temperature regulation, light regulation`)
- State (`temperature state, light intensity state, on/off state, open/close state`)
- {`Humidity, Temperature, Pressure`} MeasurementNotification

4 http://elite.polito.it/ontologies/dogont.owl

Staroch design an ontology for smart homes and related to the weather [41]. This ontology is referenced by LOV. This ontology enables to deduce if there is a need to irrigate the garden, to open the windows and when do we have to keep them shut, do we need sun protection?

They define numerous concepts related to weather sensors such as `temperature, humidity, dew point, wind speed and direction, precipitation intensity and probability, atmospheric pressure, cloud cover, solar radiation, sun's position.`

Their SWRL rules enable to deduce new information, for instance with a temperature measurement, we can infer:

- `Frost` (for an observed temperature value of below 0°C)
- `Cold` (at least 0°C and less than 10°C)
- `Below room temperature` (at least 10°C and less than 20°C)
- `Room temperature` (at least 20°C and at most 25°C)
- `Above room temperature` (more than 25°C and at most 30°C)
- and `Heat` (more than 30°C).

Riboni [38] [17] [37] [36] propose a human activity recognition ontology:

- Concepts: activity (bathing, brushing teeth, combing hair, eating, showering, sleeping), building, bus, car, carnaval party, clothing, beach, river, road, bedroom, beach umbrella
- Sensors and actuators used: Humidity, light, temperature, pressure
- Rules: temperature pressure , door status (open close), light status (high low medium off), phone status (busy, idle), water heater status (on off)

Figure 33 Part of the ontology of activities - Riboni et al. [36]

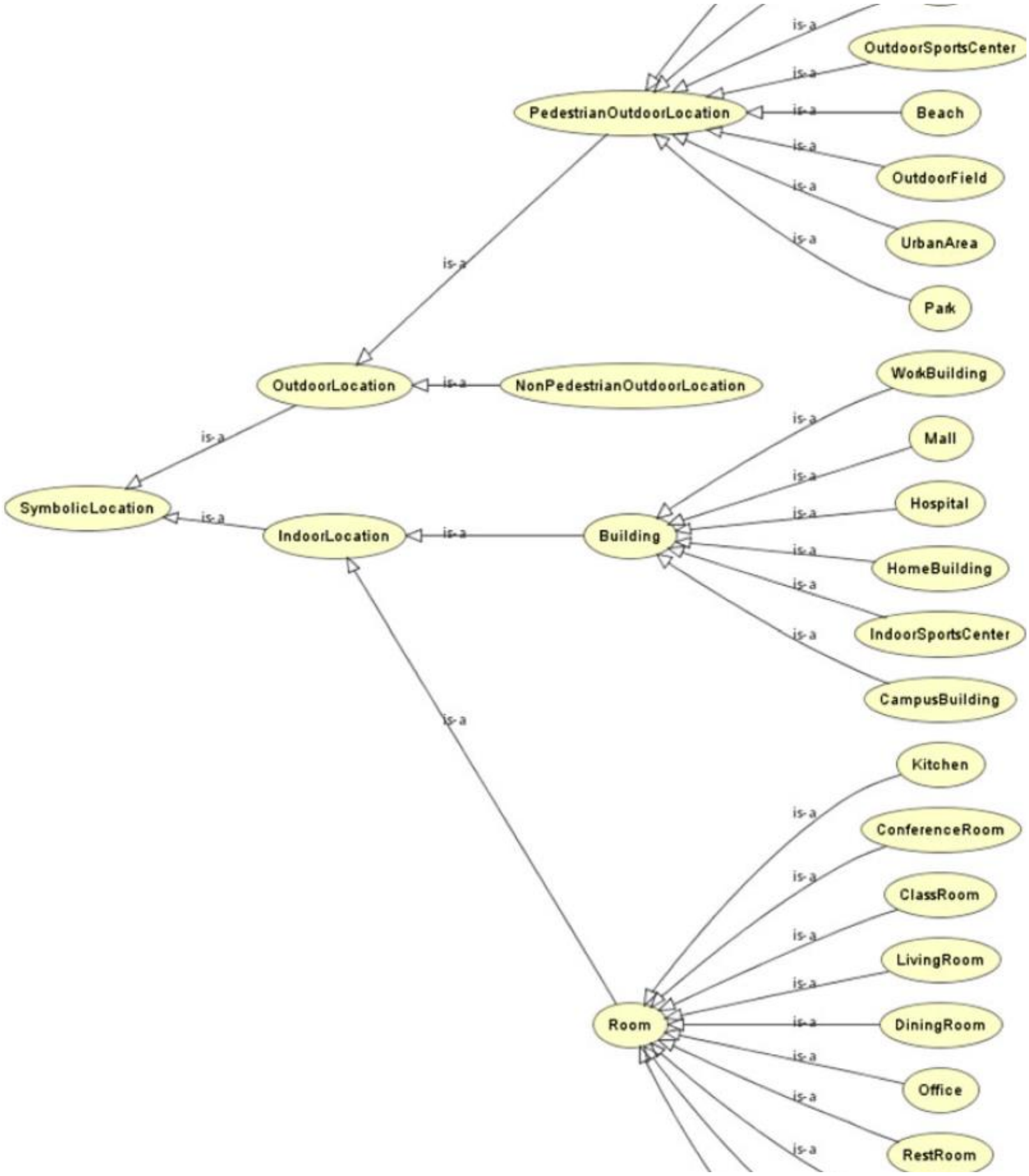


Figure 34 Part of the ontology of symbolic locations - Riboni et al. [36]

Bonsai [42]:

- Concepts: Noise, co2 level, room, air condition, light,
- Technologies used: zigbee, z-wave, W3C SSN ontology, DUL, protege editor tool

Kofler et al. [20] propose the ThinkHome ontology [35], where they describe:

- Energy: nonrenewable energy such as coal, nuclear, oil, natural gas and renewable energy like water wind, solar, wood…
- Energy providers: electric, gas, water, wood.
- Energy tariffs
- Energy facilities
- Energies properties

Their prototype propose a self-regulation of heating and cooling system tailored to schedule (nigh-time, weekends, holidays, seasons).

Wongpatikaseree et al. [49] design an ontology to detect activities in a smart home.

Wemlinger et al. [48] define the COSE ontology and numerous sensors (`binary pressure sensor, barometric pressure sensor, passive infrared sensor, gyroscope, shake sensor, accelerometer, smoke alarm, microphone, contact sensor, flow sensor`) to deduce `activities` (`cleaning, cooking, drinking, eating, making phone call, toileting, washing hands`).

Preuveneers et al. define the Codamos [33] ontology. This work is based on sensors (`Temperature, Pressure, Humididity, Lighting, Noise`) and defined the related rules such as turn on/off the lights according to the weather (cloudy, rainy) or if the person is located in the room.

Chen, Finin, Joshi and Perich worked on the SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) ontology [9] [10] [12] to describe user profiles, beliefs, desires, etc. and the COBRA architecture [7] [8] [11] to build smart meeting rooms. COBRA (Context Broker Architecture) developed by Chen, Finin et al. is a centralized architecture for context-aware systems in smart environment based on semantic web languages. This architecture does not use SWE standards. They developed EasyMeeting, an intelligent meeting room based on the COBRA architecture. They define a policy language for users to control the sharing of their information and two ontologies SOUPA and COBRA-ONT. The ontology COBRA-ONT is for modeling context in an intelligent meeting room:

- Places (a physical location: longitude, latitude, and string name). They propose AtomicPlace (a room, an hallway, stairway, restroom, parking lot) and CompoundPlace (e.g., Campus or building are comprised of rooms)
- Agents are Person (name, homepage, email address) or SoftwareAgent.
- Agent's Location can detect some inconsistencies (a person who are in the same time in a parking lot and in a room).
- Agent's Activity represents for instance a meeting (A PresentationSchedule with the start time, the end time, the presentation title etc.)

The SOUPA Ontology is split into:

- SOUPA Core which attempt to define generic vocabularies that are universal for different pervasive computing applications.
- SOUPA Extension defines additional vocabularies for supporting specific types of applications.

The Soupa[5] ontology defined by Chen et al. is composed of 11 ontologies (assertion, association, conference, contact, event, news, person, photo, project, publication, research). The person ontology redefines similar concepts without be linked to the FOAF ontology (name, firstName, middleName, lastName) and propose additional concepts such as PhDStudent, Visitor, GuestSpeaker, Professor, Student, etc. and interesting properties such as biography, relatedPublications to obtain additional information about the person.

9.2 Health Ontologies

Lafti et al. [21] design 7 ontologies[6] using Protegé:

- Equipment smart home ontology contains the description of all pieces of equipment that can be found in the habitat in order to ensure the patient safety included `Sensors` such as `motion detector, temperature, body temperature, presence detector, gas, light, blood pressure, fall detector` and `actuators (door, drawer, cupboard, window).`
- Person and medical history ontology describes the patient concept, his diseases, allergies, and the person concept including the relationships with the family. Unfortunately is not linked with well-known ontologies such as FOAF or relationships. Common concepts are `has Allergy, hasDisease, Allergy, ArterialHypertension, Diabetes, Person, Patient.`
- Task ontology recognizes activities using Bayesian networks. `Activities` describes are `Brushing, Cooking, crying, eating, reading (Book, Newspaper), Sitting, Sleeping (Bed, Sofa), Speaking, Standing, Walking, Washing (Clothes, Dishes, Face, Hands)`, etc.
- Habitat ontology describes the smart home with `Rooms` concepts such as `Bedroom, Bathroom, Dining Room, Hall, Kitchen, Living Room.`
- Software application
- Behavior
- Decision

Yao et al. [50] [51] propose the CONFlexFlow (Clinical Context based Flexible workflow) framework, design 2 ontologies (clinical context ontology and heart failure ontology) and the two kind of reasoning (rule-based and ontology-based reasoning). They use Protégé 3.4 to design the ontology, the Jess rule engine to enable SWRL reasoning, the SWRLJessTab Protégé plugin to implement rules and the Pellet reasoned to find inconsistencies and infer new instances or classes. They define 18 rules:

- Patient Evaluation Rules (PER) evaluate a patient's medical history, social background, habits, symptoms prior to a physical examination.
- Patient Diagnosis Rules (PDR) evaluate patient's signs (high blood pressure or abnormal heart rhythm) to infer symptoms such as (blood cell disorder, directly heart failure, heart disease or circulation disorder.
- Patient Treatment Rules (PTR) suggest treatment such as surgical therapy, medication or device therapy.
- Patient Prescription Checking Rules (PCR) to deal with drug interaction (allergy-drugs effects, dosage checking and insurance checking) to avoid prescription errors.

---

[5] http://ebiquity.umbc.edu/ontology/

[6] http://www.gdst.uqam.ca/Documents/Ontologies/HIT/

Hennessy et al. [18] propose two ontologies : Healthcare Semantics Lite (HSL) to represent the patient and another ontology dedicated to the medical context. The both ontologies enable to reduce the interoperability issues between medical sensors, smartphones and hospital patient record systems. They use the Schema.org, an ontology supported by Google, Yahoo, etc. The medical reading concepts defined are: `WeightScale, Temperature, Pulse, BloodPressure and Glucose`. They used the SPINMap[7] and SPIN to define rules, REST-full web services, the Amazon EC2 cloud-based server, SPARQLMotion scripts and the TopBraid semantic web tool.

Roose et al. [1] uses various sensors and actuators such as `ultrasonic water flow meter, ip camera, flush detector, light switch, door, fridge sensor, hob sensor, mixer tap, mobile phone gps` and `sound detector`. They use the Jena framework, Protégé and SWRL to deduce activities (`dressing, eating, elimination, hygiene, lie down, preparation eating`, etc.)

Lukkien, brandt [5] [23] propose an ontology for a remote patient monitoring.

Paganelli [29] [30] design an ontology to monitor and assist patient at home and a reasoning for alarm situation handling. Their work are based on biomedical en environmental sensors and define four ontologies:

- The patient-personal domain ontology to estimate patient's health status (`body temperature, heart rate frequency, pulse oxymetry, systolic and diastolic blood pressure, glycemia`). When a measured value falls out of the thresholds, the rules trigger alarms (very low, low, medium and high)
- The home domain ontology to monitor environmental parameters (`temperature, humidity`) and detect abnormal situations with the help of gas and fire detectors.
- The alarm management ontology to trigger alarm.
- The social context ontology to alert available persons (nurse, caregiver, family member)via SMS or email.

They propose two kind of reasoning:

- Ontology-based reasoning to determine class subsumption.
- User defined rule-based reasoning to make inferences over the knowledge base. For instance, they describe rules to trigger alarms and alert available people in case of the heart rate frequency is less than 40 beat/minute and systolic blood pressure is higher than 160mm/Hg.

Taboada et al. [44] define SWRL rules using the Protégé SWRLTab to reason about juvenile cataracts.

Jovic [19] define the heart failure ontology.

Zhao [52]

Ontoreachir[8] [25] defines 2039 concepts and 200 relations for the reanimation surgery domain. We link concepts related to `Disease` and blood measurements (`HypertensionArterielle`, `Hypoglycemie`).

Physicology[9] describes concepts related to blood (`Pressure, Glucose`).

---

[7] http://composing-the-semantic-web.blogspot.fr/2011/04/spinmap-sparql-based-ontology-mapping.html
[8] Search on google (filetype:owl Ontoreachir)
[9] Search on google (filetype:owl Physicology)

The registry ontology[10] defines interesting concepts related to Patient or `Person` (name, age, height, weight, sex, blood type) and numerous diagnostics. This ontology is not linked to the FOAF ontology whereas both ontologies describe a `Person` and have some properties in common (`hasName`).

10. Reference the domain knowledge

Once domain experts have designed and implemented their domain knowledge, they can share it through the Web. They can share the ontologies, datasets and rules.

10.1 Ontology catalogue

10.1.1 Linked Open Vocabularies (LOV)

The Linked Open Vocabularies[11] is a catalogue, created by the semantic web community which references more than 412 well-designed ontologies according to the semantic web best practices as depicted in the Figure 35.

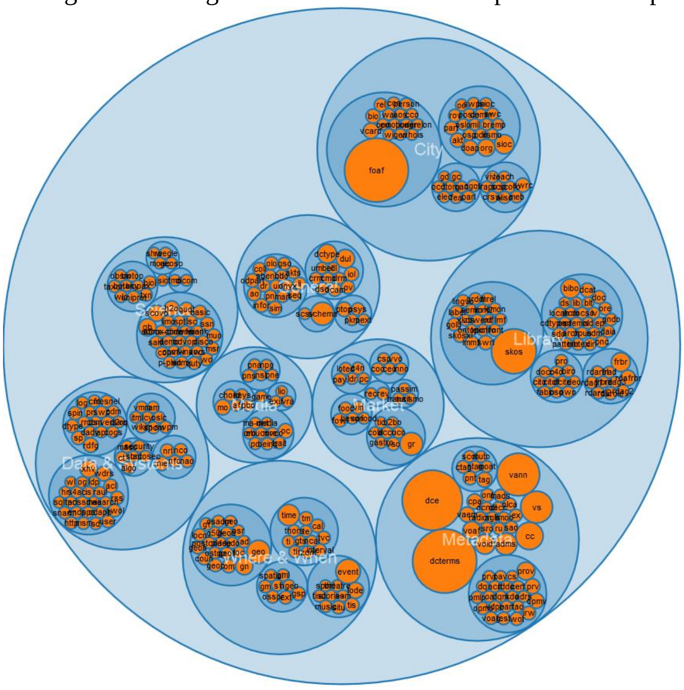


Figure 35 The Linked Open Vocabularies (LOV) catalogue

10.1.2 Linked Open Vocabularies for Internet of Things (LOV4IoT)

More than 170 **domain ontologies** have been designed by domain experts in various domains and cannot be referenced on the LOV catalogue since they do not respect the semantic web best practices. For this reason, these 170 domain ontologies have been referenced on this web site[12].

The ontologies are classified by:

- Domains such as building automation , healthcare, security, weather forecasting, intelligent transportation systems, affective science, tourism, agriculture, food, etc.

---

[10] http://ontology-for-registry-of-children-with-special-needs.googlecode.com/svn-history/r23/trunk/Registry3.4.4.owl

[11] http://lov.okfn.org/dataset/lov/

[12] http://www.sensormeasurement.appspot.com/?p=ontologies

- Date
- Ontology status as displayed in the Figure 36:
    - Colored in white: Domain experts do not answer to emails
    - Colored in red: the ontology cannot be shared for diverse reasons (lost, confidential, etc.)
    - Colored in purple: domain experts intent to share and publish the ontology soon
    - Colored in green: the ontology is published online but not according to the semantic web best practices
    - Colored in yellow: the ontology is published online and the semantic web best practices are complied with
    - Colored in orange: few of them were already published online according to the semantic web best practices

- The ontology will never be available (lost, confidential, etc.) :-(
- We are waiting the response of the authors to publish the ontology online
- Authors are publishing online the ontology (ongoing work)
- Ontology published online but the semantic web best practices are not complied with.
- Ontology published online and referenced by LOV since semantic web best practices are adopted!
- Already on LOV - No email sent

Figure 36 Color code for the ontology status

Smart Home, Building Automation, Activities of Daily Living, Ambient Assited Living ontologies

Authors	Year	Paper	Url onto	Technologies	Sensors	Rules
De paola Mail: 05/02/14, 25/02/14	2014	Book Chapter: An ontology-based autonomic system for ambient intelligence scenarios	??? Ontology URL ??? Concepts: Sensor, Actuator, Device, Closed, RoomOccupancy	SWRL, JESS	Light, Sound, pressure, temperature, humidity, door (close/open)	
Park et al Mail: 17/02/14, Response: 20/02/14	2013	Paper: A feedback-based approach to validate swrl rules for developing situation-aware software	Cannot share the code (research regulation). Concepts: fire	SWRL	temperature, humidity, c02	rule (age -> adult, fire, temp too hight, humidity too low)
Nguyen, Raspitzu et al. Mail: 24/02/14, Response: 26/02/14, 26/02/14	2013	Paper: Ontology-based office activity recognition with applications for energy savings.	??? Ontology URL ??? Concepts: working room (PC), meeting room (presentation), coffee corner (having coffee, having lunch)	Protege, Hermit, Java API	Acoustic, pressure, PIR (Passive Infrared)	
Kofler et al. - ThinkHome	2011	Paper: Thinkhome energy efficiency in future smart homes 2011 Paper: A semantic representation of energy-related information in future smart homes	Ontology URL Concepts: Weather(rain, hail, sleet, snow, thunder, sun cloud, fog) Onto + Dataset + rules	Pellet	Occupancy sensor, temperature, humidity, lighting, ventilation, solar panel, wind turbine, actuator (heating, washing machine, dishwasher, window)	Rules owl restrictions ontology

Figure 37 Ontology status in the building automation domain

Domain	Total onto	# No answer	# onto online	# onto lost	# ongoing onto	# ref by lov
Transport	26	11	6	4	5	0
Building Automation	29	10	7	3	8	1
Healthcare	34	10	12	7	5	0
Security	20	5	8	1	2	4
Tourism	26	10	10	4	1	1
Affective Science	5	1	2	0	0	2
Food, Beverage, Restaurant	22	9	9	0	3	1
Agriculture	7	5	1	1	0	0
Weather	9	2	5	0	0	2
Earthquake, pollution, environment	7	4	3	0	0	0
Total	185 (100%)	67 (36%)	63 (34%)	20 (11%)	24 (13%)	11 (6%)

Figure 38 More 184 ontology status are classified by domain

10.2 Dataset catalogue

- The DataHub[13] project proposes an easy way to get, use and share data as depicted in the Figure 39.
- The Linked Open Data search engine as depicted in the Figure 40.

[13] http://datahub.io/en/

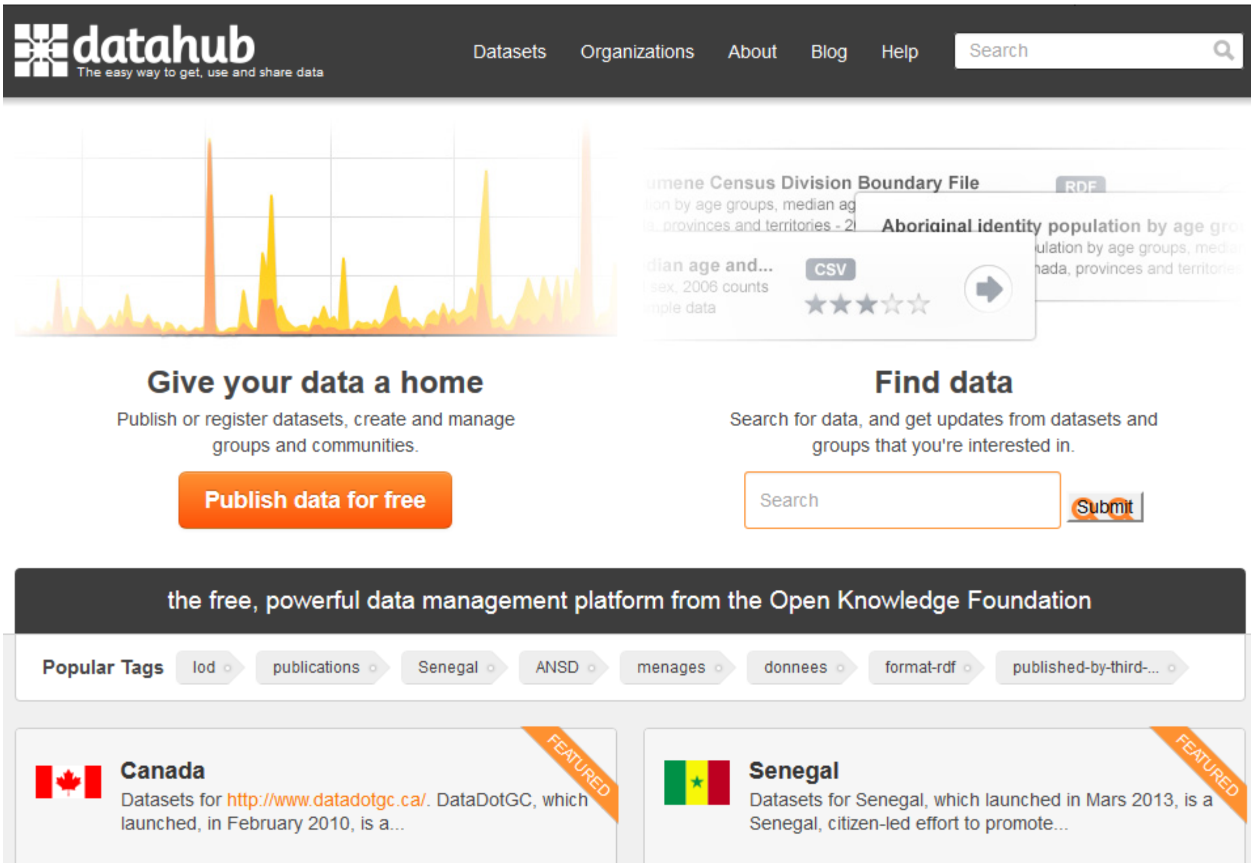


Figure 39 DataHub

Figure 40 Linked Open Data search engine

10.3 Rules Catalogue

The "Linked Open Rules", a work in progress, intents to share reuse and combine existing semantic web rules.

11. Semantic web tools

11.1 Ontology editors, semantic API or framework

- Protégé[14] is the most used ontology free editor tool to design a new ontology as depicted in the Figure 41 and proposes various plugin for ontology visualization, writting rules, etc.
- Callimachus
- TopBraid is a commercial solution to build semantic web and linked data applications
- SWOOP is a tool for creating, editing, and debugging OWL ontologies.
- Jena compatible with JAVA
- Virtuoso
- Sesame
- NeOn Toolkit
- OWL API as depicted
- OWLed2[15]

[14] http://protege.stanford.edu/
[15] http://www.doom-srl.it/index.php?option=com_content&task=view&id=20&Itemid=30&lang=en

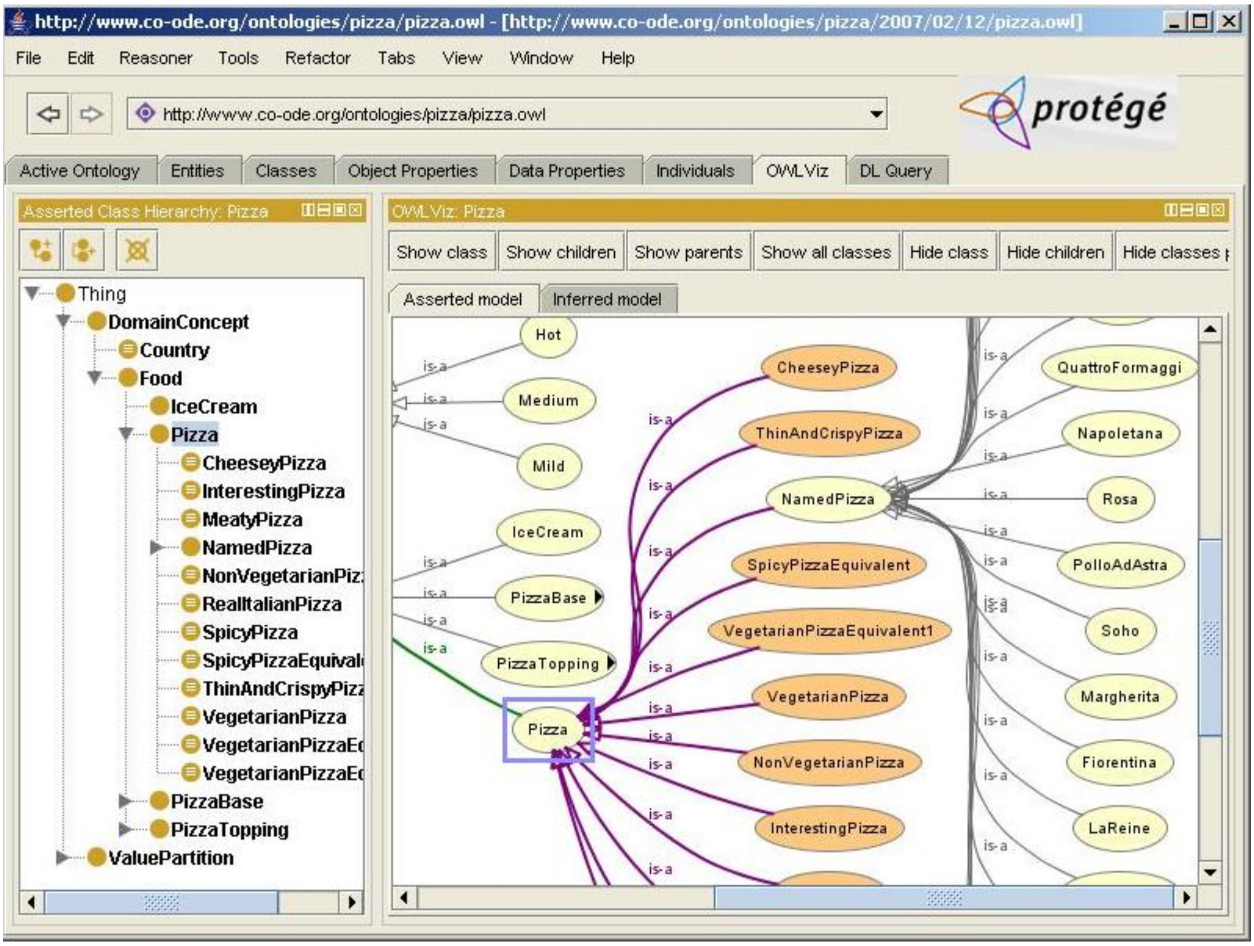


Figure 41 Protégé Editor tool

11.2 Mapping tools

- LogMap is used to link ontologies with each other
- Silk is used to link datasets with each other
- SameAS is used to link datasets with each other
- LIMES (Linked Discovery Framework for Metric Spaces)
- RiMOM
- idMash
- ObjectCoref

11.3 Linked data search search engines

- Sindice provides API which can be used by Linked Data applications.
- Watson provides API which can be used by Linked Data applications.
- Swoogle provides API which can be used by Linked Data applications.
- OpenLink Data Explorer
- SchemaCache
- SchemaWeb
- Sig.ma
- Falcons
- SWSE

11.4 Linked data browsers:

- Disco hyperdata browser
- Tabulator browser
- LinkSailor
- LOD Browser switch

11.5 Semantic Reasoner

- Jess
- Pellet is an OWL 2 reasoner for JAVA.
    - Pellet - Protege
    - Pellet - Jena
- Racer
- Kaon
- Fact++
- Hermit

11.6 Converter

- Datalift
- SenML to RDF Converter

11.7 Others

- Pubby
- Sindice Web data inspector: http://inspector.sindice.com/
- Purl
- Pachube
- URI validator: http://www.hyperthing.org/
- DSNotify informs consuming applications about changes.
- RDFa Distiller and Parser: http://www.w3.org/2007/08/pyRdfa/

12. Serialisation

12.1 Turtle

Turtle is more readable by human.

12.2 N3

12.3 Rdf/xml

Rdf/xml is widely supported by tools that consume Linked Data.

13. Annexe A: Ontology LOV metadata

Example:

```
        <owl:Ontology
rdf:about="http://securitytoolbox.appspot.com/securityAlgorithms#">
```

```
                <rdfs:comment> An ontology to describe various
cryptographic algorithms</rdfs:comment>
                <rdf:type
rdf:resource="http://purl.org/vocommons/voaf#Vocabulary"/>
                <dc:title xml:lang="en">Security Algorithms</dc:title>
                <skos:historyNote xml:lang="en">Ontology extracted from the
paper Security Ontology for Annotating Resources. [Kim et al. 2005] (See
APPENDIX D. OWL Representations of the NRL Security Ontology) Security
ontology to faciliate web service description and
discovery.</skos:historyNote>
                <dc:description xml:lang="en">An ontology to describe
various cryptographic algorithms</dc:description>
                <dcterms:source rdf:resource="http://www.dtic.mil/cgi-
bin/GetTRDoc?AD=ADA437938"/>
                <dcterms:creator>
                        <foaf:Person
rdf:about="mailto:kim@itd.nrl.navy.mil">
                        <foaf:name>Anya Kim</foaf:name>
                        </foaf:Person>
                </dcterms:creator>
                <dcterms:issued
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2005-08-
31</dcterms:issued>
                <dcterms:modified
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2014-01-
24</dcterms:modified>
                <owl:versionInfo
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.2</owl:versionInf
o>
                <vs:term_status>Finished</vs:term_status>
        <cc:license
rdf:resource="http://creativecommons.org/licenses/by/3.0/"/>

        <vann:preferredNamespacePrefix>algo</vann:preferredNamespacePrefix>

<vann:preferredNamespaceUri>http://securitytoolbox.appspot.com/securityAlgo
rithms#</vann:preferredNamespaceUri>
        </owl:Ontology>
```