**INPUT CONTRIBUTION**

<table>
<thead>
<tr>
<th>Group Name:*</th>
<th>MAS WG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:*</td>
<td>Semantic Web best practices. Semantic Web Guidelines for domain knowledge interoperability to build the Semantic Web of Things</td>
</tr>
<tr>
<td>Source:*</td>
<td>Eurecom, Amelie Gyrard, Christian Bonnet</td>
</tr>
<tr>
<td>Contact:</td>
<td>Amelie Gyrard, <a href="mailto:gyrard@eurecom.fr">gyrard@eurecom.fr</a>, Christian Bonnet, <a href="mailto:bonnet@eurecom.fr">bonnet@eurecom.fr</a></td>
</tr>
<tr>
<td>Date:*</td>
<td>2014-04-07</td>
</tr>
<tr>
<td>Abstract:*</td>
<td>This contribution proposes to describe the semantic web best practices, semantic web tools, and existing domain ontologies for uses cases (smart home and health).</td>
</tr>
<tr>
<td>Agenda Item:*</td>
<td>Tbd</td>
</tr>
<tr>
<td>Work item(s):</td>
<td>MAS</td>
</tr>
<tr>
<td>Document(s)</td>
<td>Study of Existing Abstraction &amp; Semantic Capability Enablement Technologies for consideration by oneM2M.</td>
</tr>
<tr>
<td>Impacted*</td>
<td></td>
</tr>
<tr>
<td>Intended purpose of document:*</td>
<td>☑ Information</td>
</tr>
<tr>
<td>Decision requested or recommendation:*</td>
<td>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).</td>
</tr>
</tbody>
</table>

**oneM2M IPR STATEMENT**

Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by all provisions of IPR policy of the admitting Partner Type 1 and permission that all communications and statements, oral or written, or other information disclosed or presented, and any translation or derivative thereof, may without compensation, and to the extent such participant or attendee may legally and freely grant such copyright rights, be distributed, published, and posted on oneM2M’s web site, in whole or in part, on a non-exclusive basis by oneM2M or oneM2M Partners Type 1 or their licensees or assignees, or as oneM2M SC directs.
1. Content
1. Content ............................................................................................................................... 2
2. List of Figures ...................................................................................................................... 4
2. Scope .................................................................................................................................. 5
3. References .......................................................................................................................... 5
4. Definitions, symbols, abbreviations and acronyms ............................................................ 8
   4.1 Definitions ................................................................................................................... 8
   4.2 Symbols ....................................................................................................................... 8
   4.3 Acronyms ..................................................................................................................... 8
5. Introduction ........................................................................................................................ 9
6. Semantic web guidelines .................................................................................................... 9
   6.1 Design your ontology .................................................................................................. 9
   6.2 Describe domain knowledge at least written in English ............................................. 9
   6.3 Ontology best practices ............................................................................................. 10
      6.3.1 Choose a good namespace ................................................................................ 10
      6.3.2 Publish online the ontology ............................................................................... 10
      6.3.3 Ontology URI deferencable, Content Negotiation Problem ............................. 11
      6.3.4 Reuse with existing ontologies .......................................................................... 13
      6.3.5 Ontology metadata: LOV recommendation ...................................................... 14
      6.3.6 Server-side configuration, Vapour ..................................................................... 15
      6.3.7 Provide an ontology documentation ................................................................. 17
      6.3.8 Validate with OOPS ............................................................................................ 18
      6.3.9 Validate your ontology with semantic web validators ...................................... 18
   6.4 Dataset best practices ............................................................................................... 19
7. Ontology interoperability ................................................................................................. 20
   7.1 Protégé ...................................................................................................................... 20
   7.2 OWL API ..................................................................................................................... 21
   7.3 TopBraid ...................................................................................................................... 21
8. Rules interoperability ....................................................................................................... 22
8.1 OWL rules interoperability ........................................................................................................ 23
  8.1.1 OWL restrictions ............................................................................................................ 23
  8.1.2 OWL rules with OWL API ............................................................................................... 24
  8.1.3 OWL rules with Protégé ................................................................................................ 25
  8.1.4 OWL rules with OWLed2 ............................................................................................... 26
  8.1.5 OWL rules with topBraid ............................................................................................... 26
8.2 SWRL (Semantic Web Rule Language) ............................................................................... 27
  8.2.1 Jena rules ..................................................................................................................... 27
  8.2.2 SWRL and DLSafeRule .................................................................................................. 27
8.3 SPIN (SPARQL Inferencing Notation) ............................................................................... 28
8.4 RIF (Rule Interchange Format) .......................................................................................... 29
9. Domain ontologies interesting for the OneM2M uses cases ......................................................... 29
  9.1 Building Automation Ontologies ...................................................................................... 29
  9.2 Health Ontologies ............................................................................................................... 34
10. Reference the domain knowledge ............................................................................................ 36
  10.1 Ontology catalogue ......................................................................................................... 36
      10.1.1 Linked Open Vocabularies (LOV) ............................................................................ 36
      10.1.2 Linked Open Vocabularies for Internet of Things (LOV4IoT) .................................. 36
  10.2 Dataset catalogue ............................................................................................................. 39
  10.3 Rules Catalogue ............................................................................................................... 41
11. Semantic web tools .................................................................................................................. 41
  11.1 Ontology editors, semantic API or framework ............................................................... 41
  11.2 Mapping tools .................................................................................................................. 42
  11.3 Linked data search search engines .................................................................................... 42
  11.4 Linked data browsers: ..................................................................................................... 43
  11.5 Semantic Reasoner .......................................................................................................... 43
  11.6 Converter ........................................................................................................................ 43
  11.7 Others .............................................................................................................................. 43
12. Serialisation .............................................................................................................................. 43
2. List of Figures

Figure 1 Ontology only written in Chinese, Spanish or German is not easily reusable.................9
Figure 2 Lafti et al. [21] design an health ontology both in English and French .......................10
Figure 3 The ontology does not have a good namespace ........................................................10
Figure 4 Bad practice since ontologies are in a zip file..............................................................11
Figure 5 Content negotiation problem....................................................................................11
Figure 6 The namespace and the ontology URI are not identical ...........................................12
Figure 7 The namespace and the ontology URI are identical..................................................12
Figure 8 Staroch et al. define a smart home ontology related to the weather [41]......................13
Figure 9 SWEET ontologies ......................................................................................................14
Figure 10 Ontology metadata recommended by LOV .............................................................15
Figure 11 Vapour tool failed .....................................................................................................16
Figure 12 Vapour error indicates to see the RFC 2616, section 14.30 ......................................17
Figure 13 Documentation example .........................................................................................17
Figure 14 Do not describe 2 ideas in the same concept ..........................................................18
Figure 15 RDF validator............................................................................................................18
Figure 16 The Oops tool detects errors when developing ontologies ......................................19
Figure 17 Linked Open Data Best practices .............................................................................20
Figure 18 Chien et al. [13] design a tourism ontology with Protege .........................................21
Figure 19 Ontology designed with OWL API [5] ........................................................................21
Figure 20 Lopez et al. designed an emotion ontology [24] with TopBraid..............................21
Figure 21 Bujan et al. designed a tourism ontology with TopBraid and the rdf/xml syntax [2] and not in english.....................................................................................................................22
Figure 22 Hennessy, Ray et al. designed an emotion ontology with TopBraid and the turtle syntax [18]......................................................................................................................................................22
Figure 23 Wongpatikaseree et al. [49] defines rules to infer activities.......................................23
Figure 24 The Star-city ontology [22] defines rules to infer if it is snowy ..............................24
Figure 25 The ThinkHome ontology [20] [35] defines rules to infer if it is snowy in the building automation domain. ..............................................................................................................25
Figure 26 Food tiscaly ontology[6] implemented with Protege ..............................................25
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


[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?


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

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOV</td>
<td>Linked Open Vocabularies</td>
</tr>
<tr>
<td>SWoT</td>
<td>Semantic Web of Things</td>
</tr>
<tr>
<td>SPIN</td>
<td>(SPARQL Inferencing Notation)</td>
</tr>
<tr>
<td>SPARQL</td>
<td>SPARQL Protocol and RDF Query Language</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>RDFS</td>
<td>Resource Description Framework Schema</td>
</tr>
<tr>
<td>OWL</td>
<td>Ontology Web Language</td>
</tr>
<tr>
<td>SWRL</td>
<td>Semantic Web Rule Language</td>
</tr>
<tr>
<td>RIF</td>
<td>Rule Interchange Format</td>
</tr>
</tbody>
</table>
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:
- 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.

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.
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.

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](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.

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](http://www.w3.org/Provider/Style/URI.html)
6.3.3 Ontology URI deferencable, Content Negotiation Problem

Once the ontology is published online, the ontology can be submitted to the LOV project. Frequently, domain experts encountered the problem Content Negotiation Problem as depicted in the Figure 5.

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

Figure 7: The namespace and the ontology URI are identical

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

  <prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <prefix name="rdfs" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
  <prefix name="rdf" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
</Ontology>
```

```
<rdf:RDF xml:base="http://kmi.open.ac.uk/projects/smartrcfe/ontologies/food.owl">
  <owl:Ontology rdf:about="http://kmi.open.ac.uk/projects/smartrcfe/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:Ontology>
```

Figure 6: The namespace and the ontology URI are not 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.

Some ontologies are no longer maintained but cannot be ignored. This is the case for SWEET implemented by the NASA which designs about 6000 concepts in 200 separate 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)
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 10 Ontology metadata recommended by LOV

© 2012 oneM2M Partners
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/intrusion_description_in_wsn) is not an absolute URI (see the RFC 2616, section 14.30)


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

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).

Check and Visualize your RDF documents

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:Description rdf:about="http://www.w3.org/"
  rdf:Description>
</rdf:RDF>
```

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.

<table>
<thead>
<tr>
<th>Pitfall Description</th>
<th>Cases</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results for P04: Creating unconnected ontology elements.</td>
<td>16</td>
<td>Minor</td>
</tr>
<tr>
<td>Results for P07: Merging different concepts in the same class.</td>
<td>3</td>
<td>Minor</td>
</tr>
<tr>
<td>A class is created whose identifier is referring to two or more different concepts.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This pitfall appears in the following elements:</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://schema.org/LandmarksOrHistoricalBuildings">http://schema.org/LandmarksOrHistoricalBuildings</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://schema.org/CollegeOrUniversity">http://schema.org/CollegeOrUniversity</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://schema.org/StadiumOrArena">http://schema.org/StadiumOrArena</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Results for P08: Missing annotations.</td>
<td>2366</td>
<td>Minor</td>
</tr>
<tr>
<td>Results for P11: Missing domain or range in properties.</td>
<td>528</td>
<td>Important</td>
</tr>
<tr>
<td>Results for P12: Missing equivalent properties.</td>
<td>74</td>
<td>Important</td>
</tr>
<tr>
<td>Results for P13: Missing inverse relationships.</td>
<td>1003</td>
<td>Minor</td>
</tr>
<tr>
<td>Results for P26: Misusing ontology annotations.</td>
<td>4</td>
<td>Minor</td>
</tr>
<tr>
<td>Results for P22: Using different naming criteria in the ontology.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ontology* Minor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Results for P27: Defining wrong equivalent relationships.</td>
<td>1</td>
<td>Critical</td>
</tr>
<tr>
<td>Results for P36: Missing equivalent classes.</td>
<td>16</td>
<td>Important</td>
</tr>
<tr>
<td>Results for P31: Defining wrong equivalent classes.</td>
<td>25</td>
<td>Critical</td>
</tr>
<tr>
<td>Results for P32: Several classes with the same label.</td>
<td>9</td>
<td>Minor</td>
</tr>
<tr>
<td>Results for P34: Untyped class.</td>
<td>51</td>
<td>Important</td>
</tr>
<tr>
<td>Results for P35: Untyped property.</td>
<td>30</td>
<td>Important</td>
</tr>
<tr>
<td>SUGGESTION: symmetric or transitive object properties.</td>
<td>131</td>
<td></td>
</tr>
</tbody>
</table>

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
- 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](image)

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 37 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.
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.

7.3 TopBraid

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

Figure 20 Lopez et al. designed an emotion ontology [24] with TopBraid
Figure 21 Bujan et al. designed a tourism ontology with TopBraid and the rdf/xml syntax [2] and not in English.

```xml
<owl:Ontology rdf:ID="toursxp">
  <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 rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <owl:Class rdf:ID="Alojamiento"/>
</owl:Class>
<owl:Class rdf:ID="Archivos">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <owl:Class rdf:ID="Patrimonio"/>
</owl:Class>
</owl:Class>
```

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

```turtle
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
    ] .
```

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 identical 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 23) is implemented with the OWL API.

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

Rule 3 (transport domain): Snow -> safety device ABS, ESP, and snow chains [39] (Figure 25) is implemented with OWLedZ²

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

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


© 2012 oneM2M Partners
8.1.2 OWL rules with OWL API

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

Figure 26 Food tiscaly ontology[6] implemented with Protege
8.1.4 OWL rules with OWLed2

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
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

```xml
#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) ]
```

8.2.2 SWRL and DL.SafeRule

**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.

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

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

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 CONSTRUCT (equivalent to SWRL rules)
- **SPINMap** is used by Hennessy et al. [18] in a health-based work.
- **SPARQL Motion**

```
SELECT ?user ?Goal
WHERE { ?User hasBodyFat 'I. }
CONSTRUCT {?user hasHighBodyFat ?I. }
FILTER (?I > 24). }
```

```
CONSTRUCT {?I value_of_MedicalSign ?M. }
WHERE { ?user hasHighBodyFat ?I. }
```

```
CONSTRUCT {?user hasGoal ?G. }
WHERE { ?G hasMedicalSign ?M. }
```

```
?user hasHighBodyFat ?I. }
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

---


© 2012 oneM2M Partners
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 ontology4, 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)
- **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]
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, Humidity, 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 being linked to the FOAF ontology (name, firstName, middleName, lastName) and proposes 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, it is not linked with well-known ontologies such as FOAF or relationships. Common concepts are has Allergy, has Disease, Allergy, ArterialHypertension, Diabetes, Person, Patient.
- **Task ontology** recognizes activities using Bayesian networks. Activities describe 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 kinds 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/](http://ebiquity.umbc.edu/ontology/)
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 SPINMap7 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.)


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]

Ontoreachir8 [25] defines 2039 concepts and 200 relations for the reanimation surgery domain. We link concepts related to Disease and blood measurements (HypertensionArterielle, Hypoglycemie).

Physicology9 describes concepts related to blood (Pressure, Glucose).

---

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 31.

![Figure 35 The Linked Open Vocabularies (LOV) catalogue](image)

**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.

---


\(^{11}\) [http://lov.okfn.org/dataset/lov/](http://lov.okfn.org/dataset/lov/)

- Date
- Ontology status as displayed in the Figure 34:
  - 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**

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Paper</th>
<th>U/st onto</th>
<th>Technologies</th>
<th>Sensors</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>De paola</td>
<td>2014</td>
<td>Book Chapter: An ontology-based autonomic system for ambient intelligence scenarios</td>
<td>??? Ontology URL ??? Concepts: Sensor, Actuator, Device, Closed, RoomOccupancy</td>
<td>SWRL, JESS</td>
<td>Light, Sound, pressure, temperature, humidity, door (close/open)</td>
<td></td>
</tr>
<tr>
<td>Park et al</td>
<td>2013</td>
<td>Paper: A feedback-based approach to validate swrl rules for developing situation-aware software</td>
<td>Cannot share the code (research regulation). Concepts: fire</td>
<td>SWRL</td>
<td>temperature, humidity, CO2</td>
<td>rule (age -&gt; adult, fire, temp too high, humidity too low)</td>
</tr>
<tr>
<td>Nguyen, Raszpitzu et al</td>
<td>2013</td>
<td>Paper: Ontology-based office activity recognition with applications for energy savings</td>
<td>??? Ontology URL ??? Concepts: working room (PC), meeting room (presentation), coffee corner (having coffee, having lunch)</td>
<td>Protege, Hermit, Java API</td>
<td>Acoustic, pressure, PIR (Passive Infrared)</td>
<td></td>
</tr>
</tbody>
</table>
Figure 37 Ontology status in the building automation domain
### Domain Total onto # No answer # onto online # onto lost # ongoing onto # ref by lov

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

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 37.
- The [Linked Open Data search engine](http://datahub.io/en/) as depicted in the Figure 38.

Figure 39 DataHub

Figure 40 Linked Open Data search engine
10.3 Rules Catalogue

The “Linked Open Rules”, a work in progress, intends to share reuse and combine existing semantic web rules.

11. Semantic web tools

11.1 Ontology editors, semantic API or framework

- Protége\(^\text{14}\) is the most used ontology free editor tool to design a new ontology as depicted in the Figure 39 and proposes various plugin for ontology visualization, writing 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\(^\text{15}\)

\(^{14}\) http://protege.stanford.edu/
\(^{15}\) http://www.doom-srl.it/index.php?option=com_content&task=view&id=20&Itemid=30&lang=en
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
<rdfs:comment>An ontology to describe various cryptographic algorithms</rdfs:comment>

<rdf:type>
  <dc:description xml:lang="en">An ontology to describe various cryptographic algorithms</dc:description>
  <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>
  <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>