



WoT ontology development & current status (v0.0.6)

María Poveda Villalón

Fernando Serena

Raúl García Castro

Ontology Engineering Group
Universidad Politécnica de Madrid

✉ mpoveda@fi.upm.es

🐦 @MariaPovedaV

📅 17st May, 2017

📍 Osaka F2F (remote)



Created in 1995

Directors: A. Gómez-Pérez, O. Corcho

Position: 8th in the UPM ranking (200 groups)

Research Group (30 people)

- 2 Full Professors
- 5 Associate Professors
- 3 Assistant Professors
- 7 Senior Postdocs
- 6 PhD Students
- 2 MSc and BSc Students
- 2 software engineers
- 1 system administrator
- 2 project managers

170+ Past Collaborators

50+ Past Visitors



<http://www.oeg-upm.net/>

<https://github.com/oeg-upm>



@oeg-upm

- Created in 1995
- World-wide known in the research areas
 - Ontologies
 - Semantic Web and Linked Data
 - Multilingual linked Data
 - Open Data
 - eScience
- **Projects (> 12M€)**
 - 27 EU projects (7 as coordinator)
 - 54 National Projects
 - 27 contracts with companies
 - Awards: SUR IBM Watson
- **Publications**
 - 106 journals
 - 362 International conferences and book chapters
 - 7 Books
- **Impact of publications H-index**
 - Asunción Gómez-Pérez (h:50, citations 15025)
 - Oscar Corcho García (h: 36, citations 8258)
- **Services to the Spanish community**
 - Host esDbpedia
 - Host linkeddata.es
- **Awards and Prizes**
 - Ada Byron, Fujitsu, Open data, ISWC
 - SUR Awards Watson for Tech. Watch
- **Supervision of students**
 - 28 Ph.D thesis (9 awarded best thesis prize)
 - >150 MS.C thesis and BS.C
- **Events organization**
 - 11 editions of the International Summer School on Ontological Engineering and the Semantic Web
 - > 50 WS and tutorials
- **Standardization activities**
 - >25 @ W3C, ISO, OASIS, AENOR, etc.
- **Mobility**
 - PhD students: 3-6 months abroad
 - Postdocs: 1 month every 2 years
- **Visibility**
 - Program chairs of ESWC, ISWC, KCAP, EKAW, TKE, TIA
 - Editorial board of Journals
 - Invited talks at conferences and events
 - Programme Committee presence
- **Collaboration with COM (Center Open Middleware)**





VICINTY's vision is to

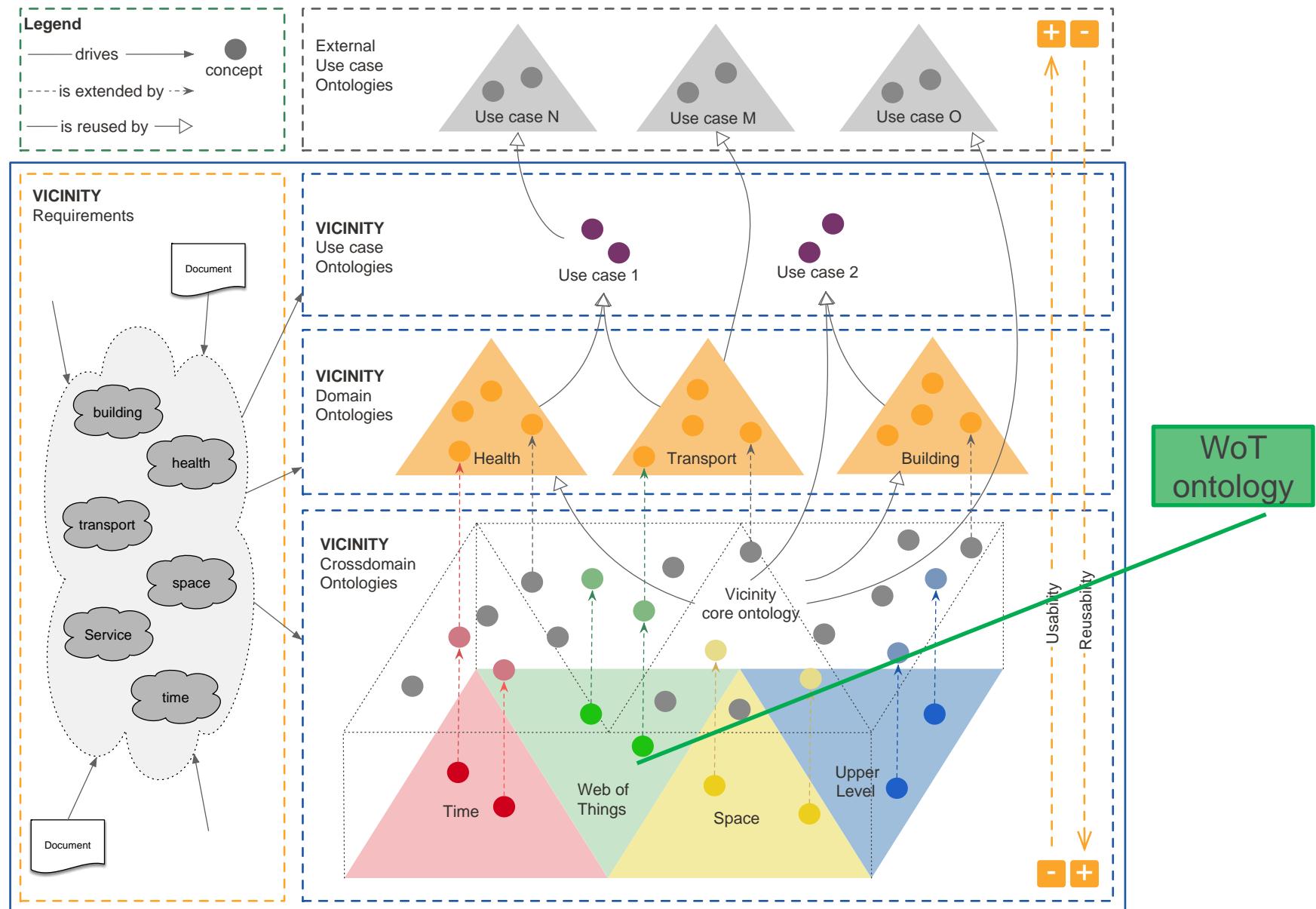
<http://vicinity2020.eu/>

- provide “**Interoperability**” as a service
- create a platform for domain-crossing, **value-added services**

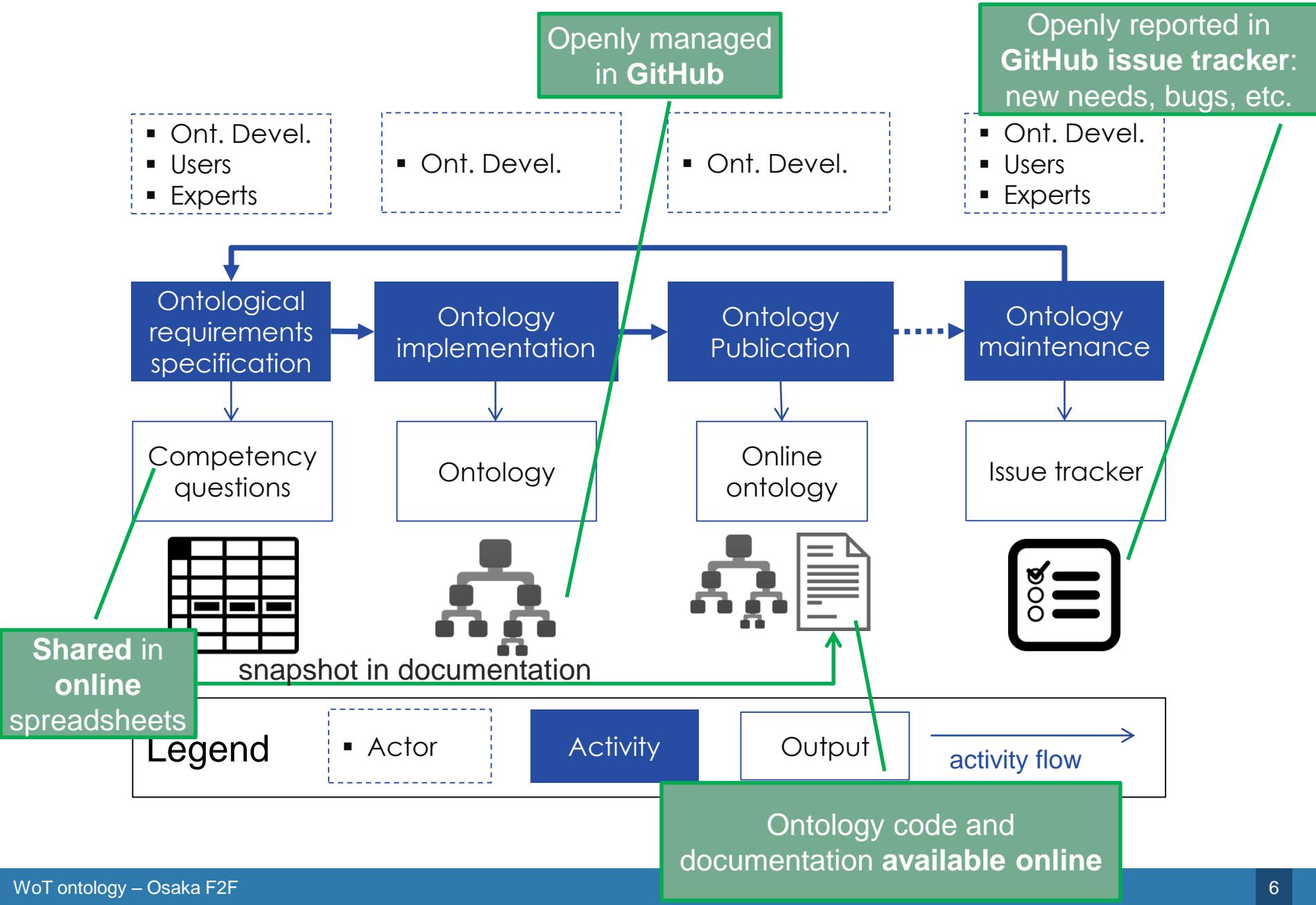
by building and demonstrating a

- **bottom-up ecosystem of decentralised interoperability** of IoT infrastructures called **virtual neighbourhood**,
- like social network for things, enabling **value added services**
 - where users can **share** the access to their smart objects **without losing the control** over them
 - where **x-domain services and buisiness models** can be established

VICINITY ontology network



Ontology development process overview



<http://w3c.github.io/wot/current-practices/wot->

■ Ont. Devel.
■ Users
■ Experts

Ontological requirements specification → Ontology implementation → Ontology

Competency questions

ontology grid

TABLE OF CONTENTS

1. Introduction
 - 1.1 Vision
 - 1.2 How to get started
2. Terminology
3. Concepts & Building Blocks
 - 3.1 WoT Interface
 - 3.1.1 Resource Model and URLs

W3C home > Mailing lists > Public > public-wot-ig@w3.org > December 2016

Towards a formal model of thing descriptions

This message [Message body] [Respond] [More options]
Related messages: [Next message] [Previous message]

From: Dave Raggett <d@w3.org>
Date: Wed, 7 Dec 2016 18:20:25 +0000
Message-ID: <D7140C17-4A50-4BC9-92D3-A6ED1399D60F@w3.org>
To: Public Web of Things IG <public-wot-ig@w3.org>

In today's Web of things Interest Group call, I was asked to provide a formal model of the RDF graphs for thing description. The question is what formalism to codify it in. One possibility could be the Shapes Constraint Language (SHACL), see: <http://www.w3.org/TR/shacl/> against a set of conditions. This could be used for validating a thing description against the following "grammar", for validating data, and for validating service compositions to check that the components are compatible. What other formalism

The following are based upon requirements derived from a broad range of use cases.

Each thing must have a thing description.
A thing description is a graph of RDF triples rooted in a given thing.
A thing description must have URI with which to access the description.
A thing may have meta-data, i.e. a set of predicate/value pairs.
A thing may have more properties, actions and events.
Each property, action and event must have a string literal as its name.
Each property, action and event may have metadata.
A property must have a data type.
A property may have a default value.
A property may be writable.
A property may require
Each property may itself have properties.
Each property, action and event may have metadata.
A data type is a core data type, or is defined in place, or by reference to a definition.
Core data types are boolean, integer, number, string, vector, thing, enum and union.
A property may be a stream.
A property may be a collection.
A collection is either ordered or unordered, but not both.
A collection is a set of string literals.
A union is an unordered set of data types.
A vector is a set of items, where each item has a string literal for its name, and a non-negative integer for its index.
A property may have constraints, which depend on its data type.
An integer or number may have a min and a max value.
A collection may have a max length.
Each action must define a request.
Each action may define a response.
A request or response may have a data type.
Each event must have a data type.
There are predefined events for signalling updates and life cycle changes.
Metadata includes comments and communication metadata.
A comment is a string literal and may be annotated with its human language.
A thing may be associated with a service.
A service provides a means to notify updates to properties and metadata.
A service provides a means to signal events, action requests and responses.
A service URI may contain variables.
A property may be a link or source, or a combination of these.
A link is a stream of samples that applications can generate.
A source is a stream of samples that applications can observe.
A stream may have a sampling rate.
A stream may have a latency.
A stream may carry date stamps.

Q: main ▲ ▼ Highlight All Match Case Whole Words 1 of 4 matches

3.2 Thing Description

The WoT Thing Description (TD) provides the [WoT Interface](#). For this, it relies on the RESTful data model. For now, [\[JSON-LD\]](#) is used as the primary vocabulary to express the capabilities of things and [Events](#). In addition, the TD provides meta-information (e.g., [name](#), [version](#), [description](#), etc.), [mediaTypes](#) (e.g., [application/json](#), [application/xml](#), etc.). [Fig. 3 Concepts of the Thing Description](#)

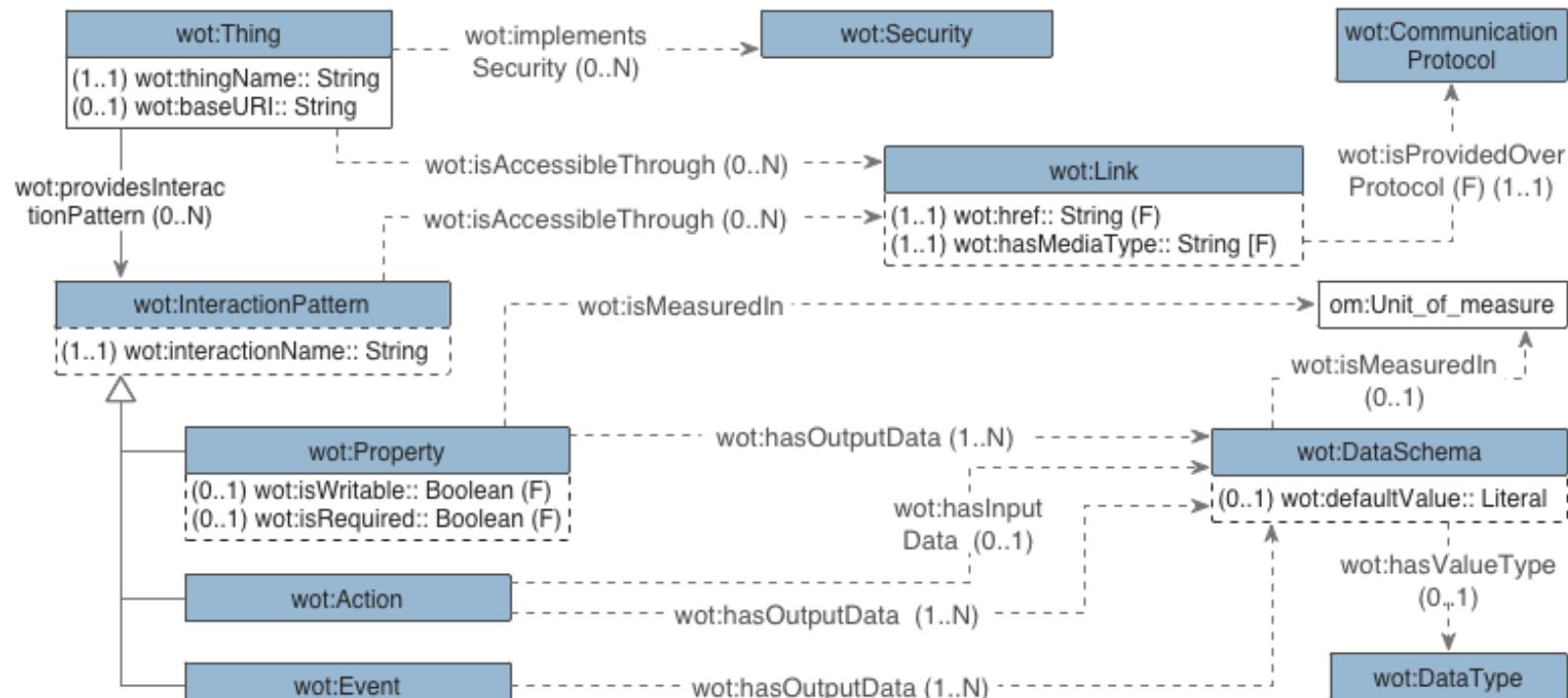
EXAMPLE 3: More Capabilities

```
{
  "@context": [
    "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    { "actuator": "http://example.org/actuator#" }
  ],
  "@type": "Thing",
  "name": "MyLEDThing",
  "base": "coap://myled.example.com:5683/",
  "security": {
    "cat": "token:jwt",
    "alg": "HS256",
    "as": "https://authority-issuing.example.org"
  },
  "interactions": [
    {
      "@type": ["Property", "actuator:onOffStatus"],
      "name": "status",
      "outputData": { "valueType": { "type": "boolean" } },
      "writable": true,
      "links": [
        {
          "href": "pwr",
          "mediaType": "application/exi"
        }
      ],
      {
        "href": "http://mytemp.example.com:8080/status",
        "mediaType": "application/json"
      }
    },
    {
      "@type": ["Action", "actuator:fadeIn"],
      "name": "fadeIn",
      "inputData": {
        "valueType": { "type": "integer" },
        "actuator:unit": "actuator:ms"
      },
      "links": [
        {
          "href": "in",
          "mediaType": "application/exi"
        }
      ],
      {
        "href": "http://mytemp.example.com:8080/in",
        "mediaType": "application/json"
      }
    }
  ]
}
```

Fig. 3 Concepts of the Thing Description

subsequent subsection will give a brief detailed explanations of the TD elements for use in production.

<https://lists.w3.org/Archives/Public/public-wot-ig/2016Dec/0016.html>



Legend:

Class

Reused Class

Class

(Card) Attribute whose domain is the attached class

Class

(Card) Attribute applicable to the attached class

For Properties

F: functional
IF: Inverse functional (only OP)
T: Transitive (only OP)

subClassOf

object property with domain and range definitions

object property applicable to the attached class

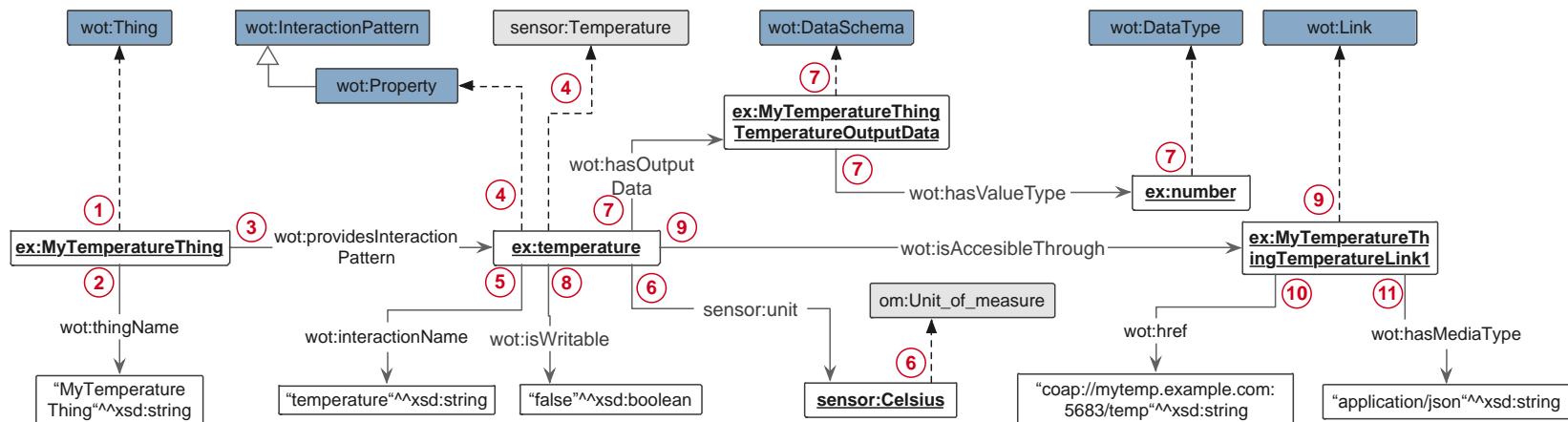
<><>stereotype<><>

Ontology:

core: <http://iot.linkeddata.es/def/core#>

Referenced ontologies:

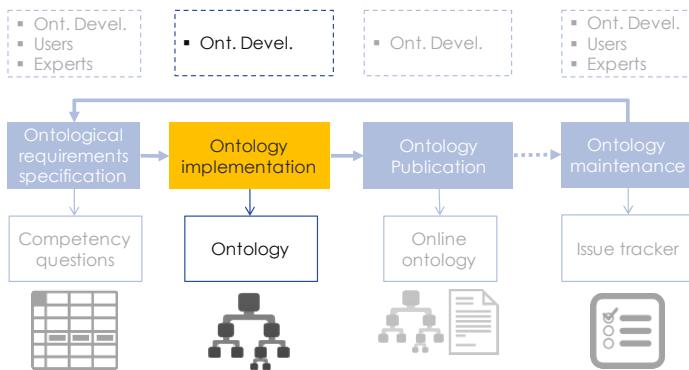
om: <http://www.wurvoc.org/vocabularies/om-1.8/>



EXAMPLE 2: Semantic Annotations

```
{
  "@context": [
    "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    { "sensor": "http://example.org/sensors#" }
  ],
  ① "@type": "Thing",
  ② "name": "MyTemperatureThing",
  ③ "interactions": [
    {
      ④ "@type": ["Property", "sensor:Temperature"],
      ⑤ "name": "temperature",
      ⑥ "sensor:unit": "sensor:Celsius",
      ⑦ "outputData": {"valueType": { "type": "number" }},
      ⑧ "writable": false,
      ⑨ "links": [
        ⑩ { "href" : "coap://mytemp.example.com:5683/", "mediaType": "application/json" }
      ]
    }
  ]
}
```

WoT Ontology implementation



```
Close Live Find Advanced Find
1 @prefix : <http://iot.linkeddata.es/def/wot#> .
2 @prefix dc: <http://purl.org/dc/elements/1.1/> .
3 @prefix vs: <http://www.w3.org/2003/06/sw-vocab-status/ns#> .
4 @prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
5 @prefix owl: <http://www.w3.org/2002/07/owl#> .
6 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
7 @prefix wot: <http://www.w3.org/2011/01/wot#> .
8 @prefix xml: <http://www.w3.org/XML/1998/namespace> .
9 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
10 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
11 @prefix prov: <http://www.w3.org/ns/prov#> .
12 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
13 @prefix vann: <http://purl.org/vocab/vann/> .
14 @prefix saref: <https://w3id.org/saref#> .
15 @prefix dcterms: <http://purl.org/dc/terms/> .
16 @base <http://iot.linkeddata.es/def/wot#> .
17
18 <http://iot.linkeddata.es/def/wot#> rdf:type owl:Ontology ;
19   dcterms:creator <http://purl.org/net/mpoveda> ;
20   dc:title "VICINITY model for Web of Things"@en ;
21   dcterms:creator <http://www.garcia-castro.com/foaf.rdf#me> ;
22   vann:preferredNamespacePrefix "wot" ;
23   dcterms:license <http://purl.org/NET/rdflicense/cc-by4.0> ;
24   owl:versionInfo "1.0" ;
25   vann:preferredNamespaceUri "http://iot.linkeddata.es/def/wot#" ;
26
```

This screenshot shows the Ontology editor interface. The top navigation bar includes tabs for Annotation Properties, Active Ontology, Class hierarchy, Annotations, Usage, Classes, Object Properties, code, rules, and Data Properties. The main area displays the `Annotations: ThingDescription` class. It lists annotations such as `rdfs:label` [language: en] and `rdfs:comment` [language: en]. The description states: "A digital representation encapsulates a physical object accessible via Web services". Below this, there are sections for Description: ThingDescription, SubClass Of, Instances, Target for Key, Disjoint With, and Disjoint Union Of. On the left sidebar, the class hierarchy is shown under the `owl:Thing` root, including categories like CommunicationProtocol, DataFormat, DataType, DigitalRepresentation, Ecosystem, Endpoint, Entity, InteractionPattern, Action, Event, Property, MediaType, Service, ThingDescription, ThingEcosystemDescription, UnitOfMeasure, WebResource, and WoTInterface.

Tool:
Ontology editor

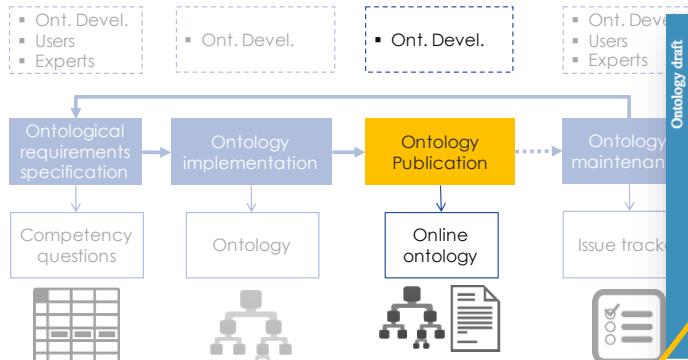
Output:
Ontology
code

This screenshot shows a GitHub repository page for `mariapoveda / vicinity-ontology-wot`. The repository description is "Repository for collaborative edition of the VICINITY module for the Web Of Things domain". It has 31 commits, 1 branch, 1 release, and 3 contributors. The repository was last updated 4 days ago. The code file `wot.ttl` is visible, showing the ontology code generated from the editor. A link to the `vicinity-ontology-wot` repository is also present.

Tool: GitHub repository
<https://github.com/mariapoveda/vicinity-ontology-wot>

WoT ontology publication (so far)

<http://iot.linkeddata.es/def/wot/>



WoT ontology

This version: <http://iot.linkeddata.es/def/wot>

Revision: 0.0.1

Authors: [Maria Poveda Villalón](#)
[Raúl García Castro](#)

Download serialization:

Format: [N Triples](#) Format: [RDF/XML](#) Format: [TTL](#)

License: [License](#) Creative Commons CC BY

Provenance of this page

Abstract

This ontology aims to model the Web of Things domain according to the w3c Interest Group (<http://w3c.github.io/wot/>)

Table of contents

space declarations
w
tion
ce for WoT classes, properti
is
Properties
Properties
ments

o model the Web of Things c

e declarations

o model the Web of Things c

Output: Machine oriented code

```
1 @prefix : <http://iot.linkeddata.es/def/wot#> .  
2 @prefix dc: <http://purl.org/dc/elements/1.1/> .  
3 @prefix vs: <http://www.w3.org/2003/06/sw-vocab-status/ns#> .  
4 @prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .  
5 @prefix owl: <http://www.w3.org/2002/07/owl#> .  
6 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
7 @prefix wot: <http://xmlns.com/wot/0.1/> .  
8 @prefix xml: <http://www.w3.org/XML/1998/namespace> .  
9 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
10 @prefix foaf: <http://xmlns.com/foaf/0.1/> .  
11 @prefix prov: <http://www.w3.org/ns/prov#> .  
12 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
13 @prefix vann: <http://purl.org/vocab/vann#> .  
14 @prefix soref: <http://www3id.org/soref#> .  
15 @prefix dcterms: <http://purl.org/dc/terms#> .  
16 @base <http://iot.linkeddata.es/def/wot#> .  
17  
18 <http://iot.linkeddata.es/def/wot#> rdf:type owl:Ontology ;  
19 dcterms:creator <http://purl.org/net/mpoveda#> ;  
20 dc:title "Vicinity model for Web of Things"@en ;  
21 dcterms:creator <http://www.garcia-castro.com/foaf.rdf#me#> ;  
22 vann:preferredNamespacePrefix "wot" ;  
23 dcterms:license <http://purl.org/NET/rdflicense/cc-by4.0#> ;  
24 owl:versionInfo "1.0" ;  
25 vann:preferredNamespaceURI "http://iot.linkeddata.es/def/wot" ;  
26 rdfs:comment "This ontology aims to model the Web of Things domain according to the w3c Interest Group  
(http://w3c.github.io/wot/)"@en ;  
27 dc:publisher <http://www.oeg-upm.net/> .  
28  
29 ##### Annotation properties #####  
30 # Annotation properties  
31 #####  
32  
33 ## http://www.w3.org/2003/06/sw-vocab-status/ns#term\_status  
34 vs:term_status rdf:type owl:AnnotationProperty .  
35  
36  
37 ##### Object Properties #####  
38 # Object Properties  
39 #####  
40  
41 ## http://iot.linkeddata.es/def/wot#describes  
42 <http://iot.linkeddata.es/def/wot#describes> rdf:type owl:ObjectProperty ;  
43 owl:inverseOf <http://iot.linkeddata.es/def/wot#isDescribedBy> ;  
44 rdfs:label "describesThing"@en .  
45
```

No project dependencies.
Proposal: <http://www.w3.org/ns/td#>

Output: Human oriented documentation

2. WoT: Overview

Classes

Action	Communication protocol	DataFormat	DataType	Digital representation	Ecosystem	Endpoint	Entity	Event	Interaction	MediaType	Physical entity	Property
Service	Thing description											

Object Properties

DescribedThing	hasComponent	hasInputData	hasMediaType	hasOutputData	hasValueType	isProvidedOver	isAccessibleThrough	isComponentOf	isDescribedBy
isMeasuredIn	providedBy	providesInteractionPattern							

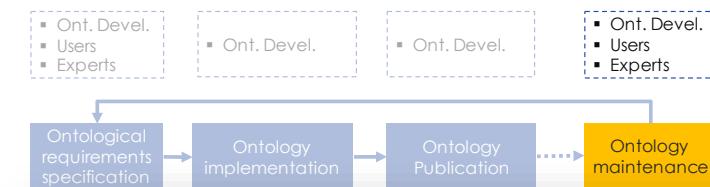
Data Properties

baseURI	defaultValue	entityName	href	interactionName	isRequired	isWritable	name

3. WoT: Description

Figure 1 provides a general overview of the current status of the WoT ontology. It is worth noting that some elements might be deleted and the model is not final as some aspects of the WoT current practices have not been addressed yet, like the security aspects.





Tool: GitHub issue tracker

<https://github.com/mariapoveda/vicinity-ontology-wot>

The screenshot illustrates the GitHub issue tracker interface for the `mariapoveda / vicinity-ontology-wot` repository. On the left, a list of open issues is displayed, with one specific issue highlighted by a yellow arrow. The issue, titled "WoT1 terminology doubt" (#4), was opened by `mariapoveda` on Nov 29, 2016. The detailed view on the right shows the initial comment from `mariapoveda` asking about the term "Thing" in the context of the Web Thing. `fserena` responds, stating that "Entity" fits the definition. Subsequent comments show `mariapoveda` adding commits related to the issue.

- **Done**

- Remove
 - Physical thing, Virtual thing, Relative endpoint, Thing description ecosystem (VICINITY concept),
 - Rename
 - Data schema (data format), Link (endpoint),
 - Add
 - Security (not described yet)
 - Changed model
 - Default value (Property → DataSchema), Media type (concept → datatype), added cardinalities
- Next steps: Currently discussing
 - Need of communication protocol?
 - Model types?
 - Coupling IG JSON model with ontology conceptual model?
 - Rename properties?
 - Not following OE good practices vs space reducing in TD?
 - Security

Thanks for your attention



WoT ontology development & current status (v0.0.6)

María Poveda Villalón

Fernando Serena

Raúl García Castro

Ontology Engineering Group
Universidad Politécnica de Madrid

 mpoveda@fi.upm.es

 @MariaPovedaV

 17st May, 2017

 Osaka F2F (remote)

