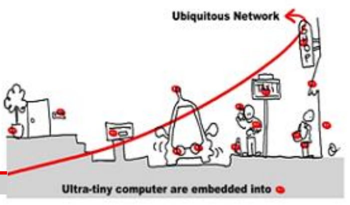
Tutorial: SWoT (Semantic Web of Things)



1 Introduction

The last decade achievements in computer hardware miniaturization and power consumption reduction has permitted the multiplication of connected devices integrated in everyday life physical objects (chair, table, lamp, etc...) and physical environments (house, building, vehicle, etc...). These devices implement resources interacting with objects (actuator) and/or gathering data (sensor) about themselves, the objects or the environment [1]. Access to these resources is achieved through services exposing their interfaces and allowing communication with the digital world. Widely deployed in so called ambient environments [2], these devices and services are selected by ambient applications (service matchmaking) that make them work in concert to assist users in several distinct domains (healthcare, smart houses, etc...). This cooperation requires a strong interoperability between devices, firstly achieved by allowing them to communicate. Although work on communication protocols (IoT, Internet of Things) tries to provide a solution to the technological heterogeneity issue, it is still challenging due to the large number of initiatives in this field [3]. Among all the possible solutions, web-services based approach (WoT, Web of Things) is now widely accepted [4].

With this hypothesis, and going a step further, the semantic heterogeneity issue inherent to the large number of heterogeneous devices and services present in the environment targeting multiple domains (smart homes, smart cities, building automation, healthcare, etc...) has to be addressed. This heterogeneity is problematic for ubiquitous systems to select, among all the available devices and services available in the environment, the most relevant ones to achieve a given functionality. On that front, semantic web technologies can be leveraged to enrich devices and services with semantic annotations used to qualify it (SWoT, Semantic Web of Things) hereby helping the selection mechanism to increase the relevancy of the selected devices and services (well beyond selection rules based on regular expressions).

2 Resources

This tutorial extends the Semantic Web of Things (SWoT) lecture available at http://www.tigli.fr/lib/exe/fetch.php?media=cours:lecture_6_swot.pdf. In addition to this lecture, interested students may have a look at this thesis (https://drive.google.com/open?id=0B6t-5TDyw60ha0hQc2hIQkd1WFk) and its associated research paper [5].

3 Tools for the tutorial

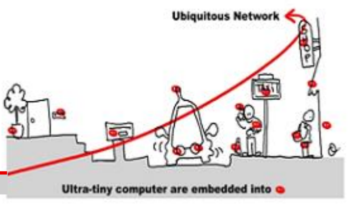
3.1 Ontology editor and knowledge management system

It exists many commercial and non-commercial ontology editors over the internet (e.g. https://www.w3.org/wiki/Ontology_editors). For this tutorial we will use Protégé, a free open source ontology editor and knowledge management system being developed at Stanford University. The tool (platform independent) can be downloaded from http://protege.stanford.edu/. It also comes with a plugin system allowing it to be extended with many interesting features.

Once installed on your computer, launch protégé. You will be requested to select the plugins to be installed (note that plugins can still be installed at any time from `File/Check for plugins…`). For this tutorial, there is no need for installing any additional plugins, default installed ones will do the job.

J.-Y. Tigli, G.Rocher
Université de Nice – Sophia Antipolis
930, Route des Colles – B.P. 145 - 06903 Sophia Antipolis Cedex – France
Tél : +33 (0)4 92 96 50 50 – Fax : +33 (0)4 92 96 50 55

Tutorial: SWoT (Semantic Web of Things)



3.2 Ontology graphical visualization tool

Although not mandatory, we will use a web-based graphical visualization tool named `WebVOWL`. The tool is accessible from http://vowl.visualdataweb.org/webvowl.html (note that a Protégé plugin is also available for this tool but is less mature than the web version. http://vowl.visualdataweb.org/protegevowl.html). Protégé comes with a pre-installed plugin to visualize ontologies (`OntoGraf`) but this tool is less user-friendly than `WebVOWL`.

4 A very basic upper ontology to start with…

Load the ontology `Base.owl` provided with this tutorial. This very simple ontology will be used as a basis for describing connected objects in the remaining parts of this tutorial.

Exercice#1
Examine the concepts and the properties defined in the ontology and their relationships. For this purpose, have a look at the ontology metrics (tab "`Active Ontology`"), the class hierarchy (tab "`Entities/Class hierarchy`"), and the object and data properties (note that missing tabs can be enabled from `Window/Tabs`).

Exercice#2
Visualize the ontology using `WebVOWL` and/or `OntoGraf` (from `Window/Tabs` one can enable `OntoGraf` tab).
Solution

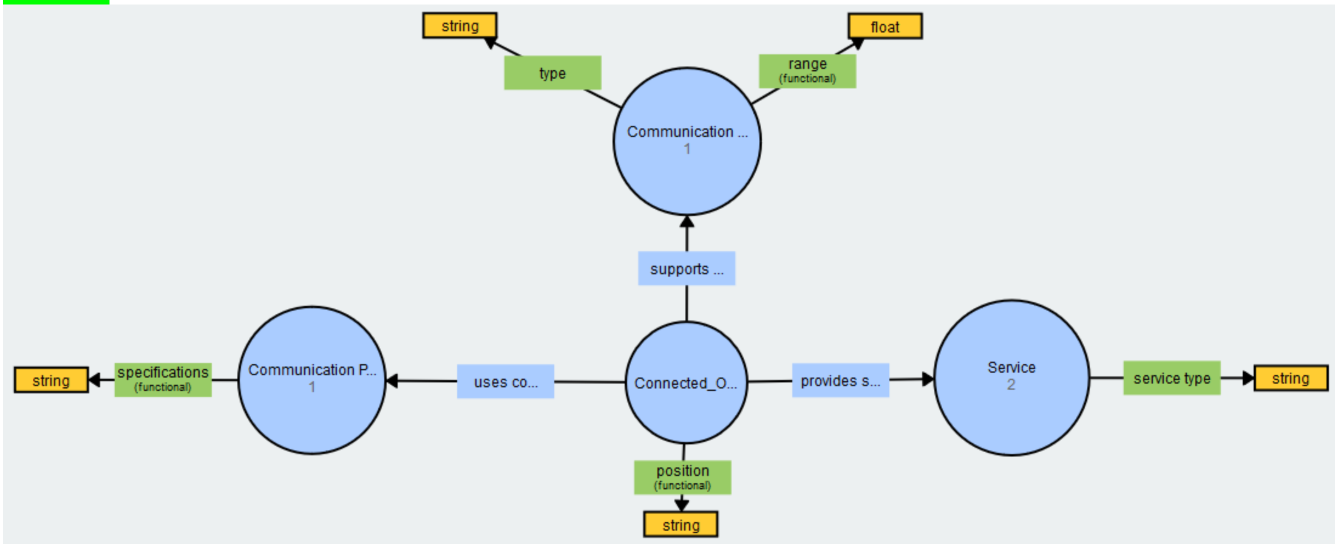


Figure 1 : A very basic ontology that can be used to describe connected objects

Connected objects are usually self-described with annotations containing their semantic description. As soon as an instance of a connected object is discovered in the environment, the annotations are published to the ubiquitous system. Once gathered by the system, the semantic descriptions are stored in a knowledge base. A knowledge base contains all the concepts and properties (aka terminology, TBOX) and all the instances (aka assertions, ABOX). The content of a knowledge base forms an ontology containing the terminology and the assertions from which reasoning and querying can be done.

Protégé embeds a knowledge base from which SPARQL queries can be executed (to do so, open the SPARQL query tab from `Window/Tabs`) and a reasoning engine that can be used to infer implicit knowledge from explicit knowledge.

Exercice#3

For the time being, we have loaded `Base.owl` ontology in the knowledge base. Are there any instances of connected objects recorded in the knowledge base?

Solution

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

PREFIX my_onto:<http://www.owl-ontologies.com/Base.owl#>

SELECT ?connected_objects
WHERE
{
   ?connected_objects rdf:type ?type .
   ?type rdfs:subClassOf* my_onto:Connected_Object
}
```

`Base.owl` is actually an upper ontology providing a vocabulary from which devices annotations can be built to semantically describe the devices and their services.

5 Adding some connected objects in the knowledge base

Let’s assume that three smart chairs have been added in the environment. The instances of these chairs are described through semantic annotations relying on the vocabulary described in the upper ontology `Base.owl`. The annotations also extend the terminology with new concepts. Once the chairs are discovered, their annotations are merged with the content of the knowledge base. Ontology *merging* is the process of generating a single, coherent ontology from two or more existing and different ontologies related to the same (or very similar) domain (See Figure 2).

Tutorial: SWoT (Semantic Web of Things)

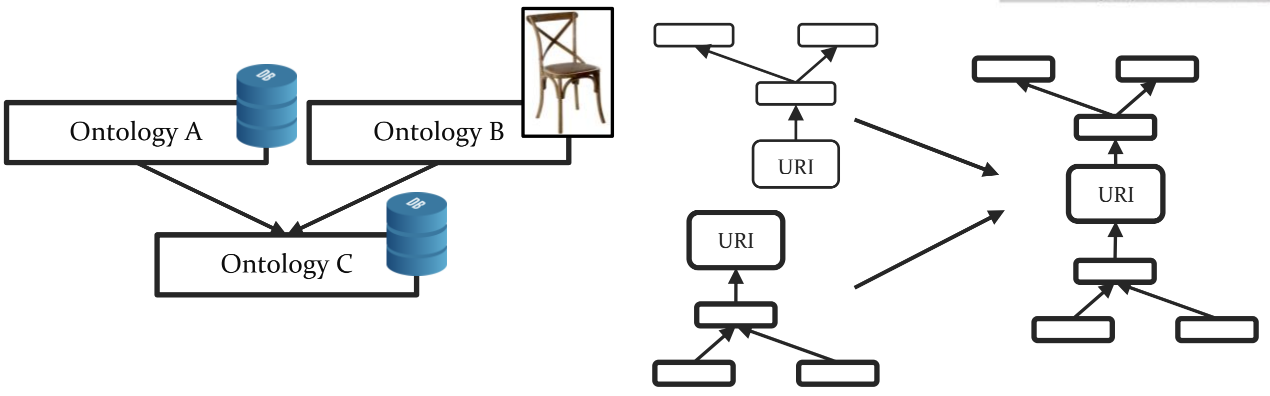


Figure 2: The content of the knowledge base (Ontology A) is merged with the content of the annotations provided by the chairs (Ontology B). The content of the knowledge is the result of the merge process (Ontology C)

Exercice#4
Load the file `SmartChair.owl` provided with this tutorial. It corresponds to the dump of the knowledge base after the chairs have been discovered and the content of their annotations merged in the knowledge base. What are the concepts added with the semantic descriptions of the newly added chairs?

Exercice#5
What are the names, the position values and the services provided by each instance of the chair discovered?

Exercice#6
Execute a SPARQL query to get all the instances of the connected objects now recorded in the knowledge base.

Based on this knowledge, one can select an instance of a chair based on its geographic position, the service it provides, the communication protocol it uses, etc… This is where semantic descriptions are powerful compared to purely syntactic descriptions.

Exercice#7
Execute a SPARQL query to get all the instances of the smart chairs in the office.
Solution

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

PREFIX my_onto:<http://www.owl-ontologies.com/SmartChair.owl#>

SELECT ?chair
```

Tutorial: SWoT (Semantic Web of Things)

```
WHERE
{
   ?chair rdf:type my_onto:Smart_Chair .
   ?chair my_onto:position "Office"^^xsd:string
}
```

Exercice#8
Execute a SPARQL query to get all the instances of the connected objects in the kitchen providing the service ‘`Buzzer`’.
Solution

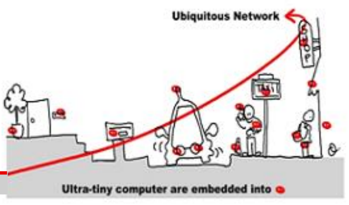
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX my_onto:<http://www.owl-ontologies.com/SmartChair.owl#>

SELECT ?connected_object ?position ?service
WHERE
{
   ?x rdfs:subClassOf* my_onto:Connected_Object .
   ?connected_object rdf:type ?x .
   ?connected_object my_onto:position ?position .
   ?connected_object my_onto:provides_service ?service
   FILTER(regex(str(?service),"Buzzer") && ?position = "Kitchen")
}
```

6 Adding a connected object in the knowledge base relying on heterogeneous knowledge description

Let’s assume now that a smart stool has been added in the environment. The instance of the stool is described through semantic annotations partially relying on the vocabulary described in the upper ontology `Base.owl`. The annotations also extend the terminology with new concepts, different from the previously defined ones. Once the stool has been discovered, its annotations are merged with the knowledge base.

Exercice#9
Load the file `SmartStool.owl` provided with this tutorial. It corresponds to the dump of the knowledge base after the stool has been discovered and the content of its annotations merged in the knowledge base. What are the concepts added with the semantic descriptions of the newly added stool?
Solution (see Figure 3)

Tutorial: SWoT (Semantic Web of Things)



Exercice#10

What is the name of the instance of the stool, its position value and the services it provides?

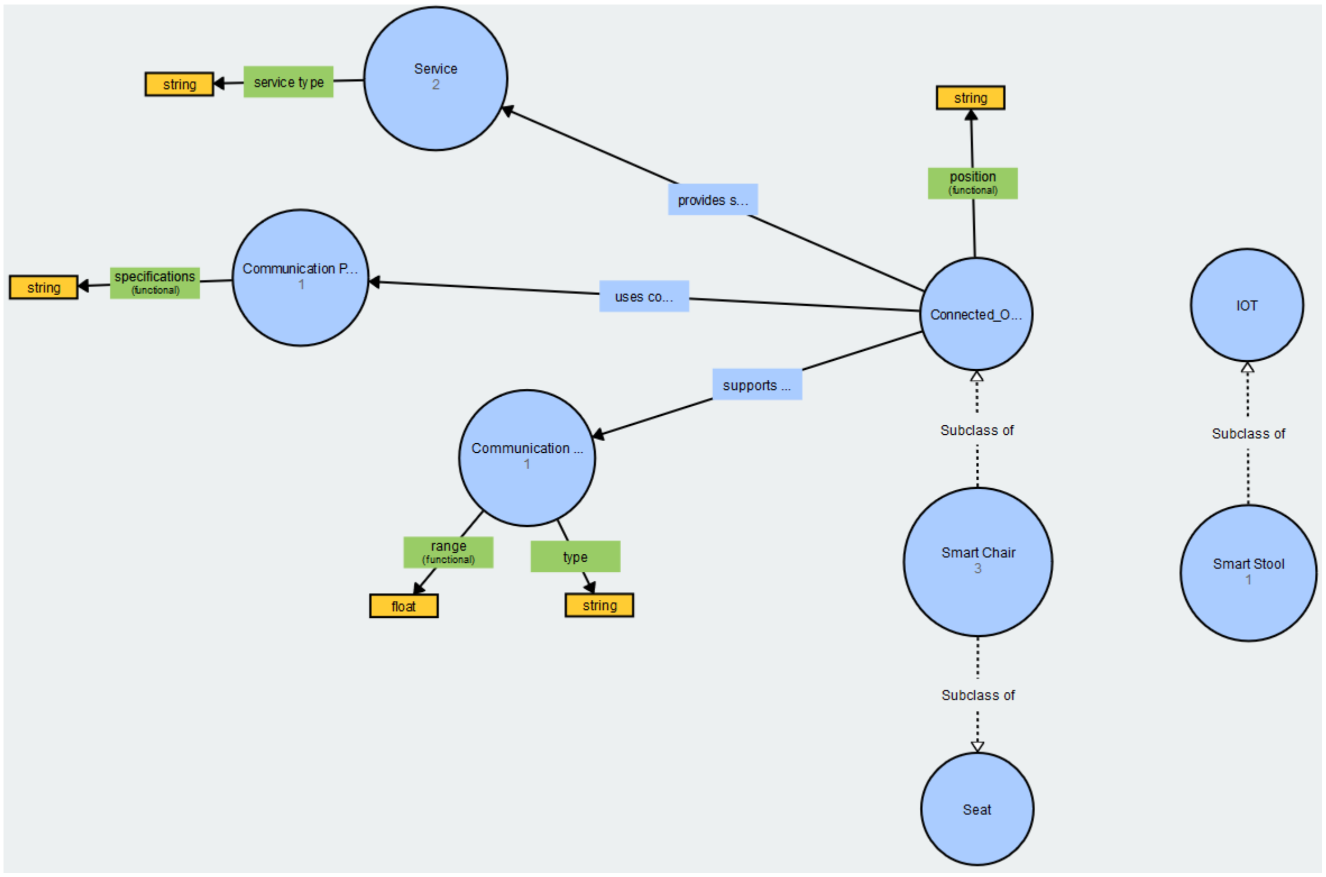


Figure 3 : The stool semantic annotations are merged with the knowledge base content. The Stool annotations define new concepts ('Smart Stool' and 'IOT') that were not previously defined in the knowledge base.

Exercise#11

Execute a SPARQL query to get all the instances of the connected objects in the kitchen providing the service 'Buzzer'. What do you remark?

Solution

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

PREFIX my_onto:<http://www.owl-ontologies.com/SmartStool.owl#>

SELECT ?connected_object ?position ?service
WHERE
{
 ?x rdfs:subClassOf* my_onto:Connected_Object .
 ?connected_object rdf:type ?x .
 ?connected_object my_onto:position ?position .
 ?connected_object my_onto:provides_service ?service
 FILTER(regex(str(?service),"Buzzer") && ?position = "Kitchen")
}
```

The stool is not listed although it provides the service Buzzer and is located in the kitchen.

7 Alignments

Ontology *alignment* process takes two or more input ontologies and produces a set of correspondences between concepts that match *semantically* with each other (Ontology matching is the process of discovering **similarities** (∈ ℝ) between two ontologies). These matches are also called mappings. Similarities can be syntactic, lexical or structural. Ontology alignment is applied if the sources become consistent with each other (describe how the concepts in the different ontologies are logically related) but are kept unchanged (without changing the meaning in the original ontologies).

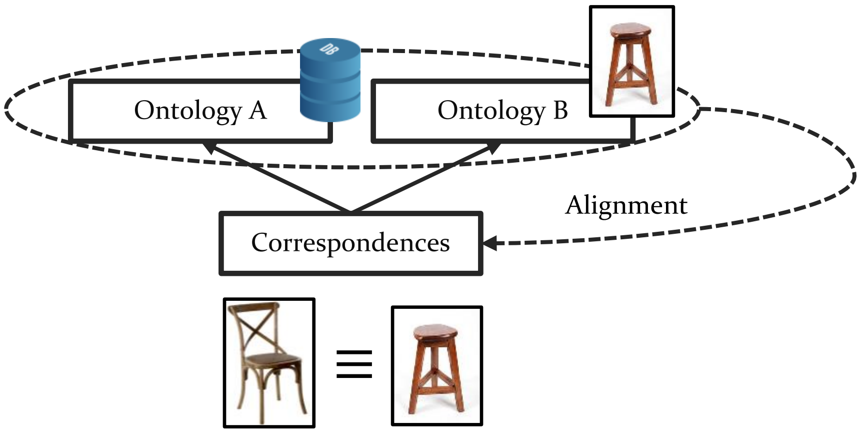


Figure 4 : Example of ontology matching. The knowledge base content (Ontology A) is aligned with the stool ontology (Ontology B). The resulting alignment is stored in "Correspondences" whose content can be further merged with the knowledge base.

Exercice#12

Load the file `Alignment.owl` provided with this tutorial. It corresponds to the dump of the knowledge base after the stool has been discovered and the content of its annotations merged in the knowledge base. It also contains the correspondences issued from an alignment process. Examine the ontology. What do you remark?

Solution (See Figure 5)

Exercise#13

Execute a SPARQL query to get all the instances of connected objects in the kitchen providing the service 'Buzzer'. What do you remark now?

Solution

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

PREFIX my_onto:<http://www.owl-ontologies.com/Alignment.owl#>
```

Tutorial: SWoT (Semantic Web of Things)

```
SELECT ?connected_object ?position ?service
WHERE
{
 ?x rdfs:subClassOf* my_onto:Connected_Object .
 ?connected_object rdf:type ?x .
 ?connected_object my_onto:position ?position .
 ?connected_object my_onto:provides_service ?service
 FILTER(regex(str(?service),"Buzzer") && ?position = "Kitchen")
}
```

Although we do expect to get "`Smart Stool`" instance listed in the result (`IOT` and `Connected_Object` concepts being equivalent), only "`Smart Chair`" instance is listed.

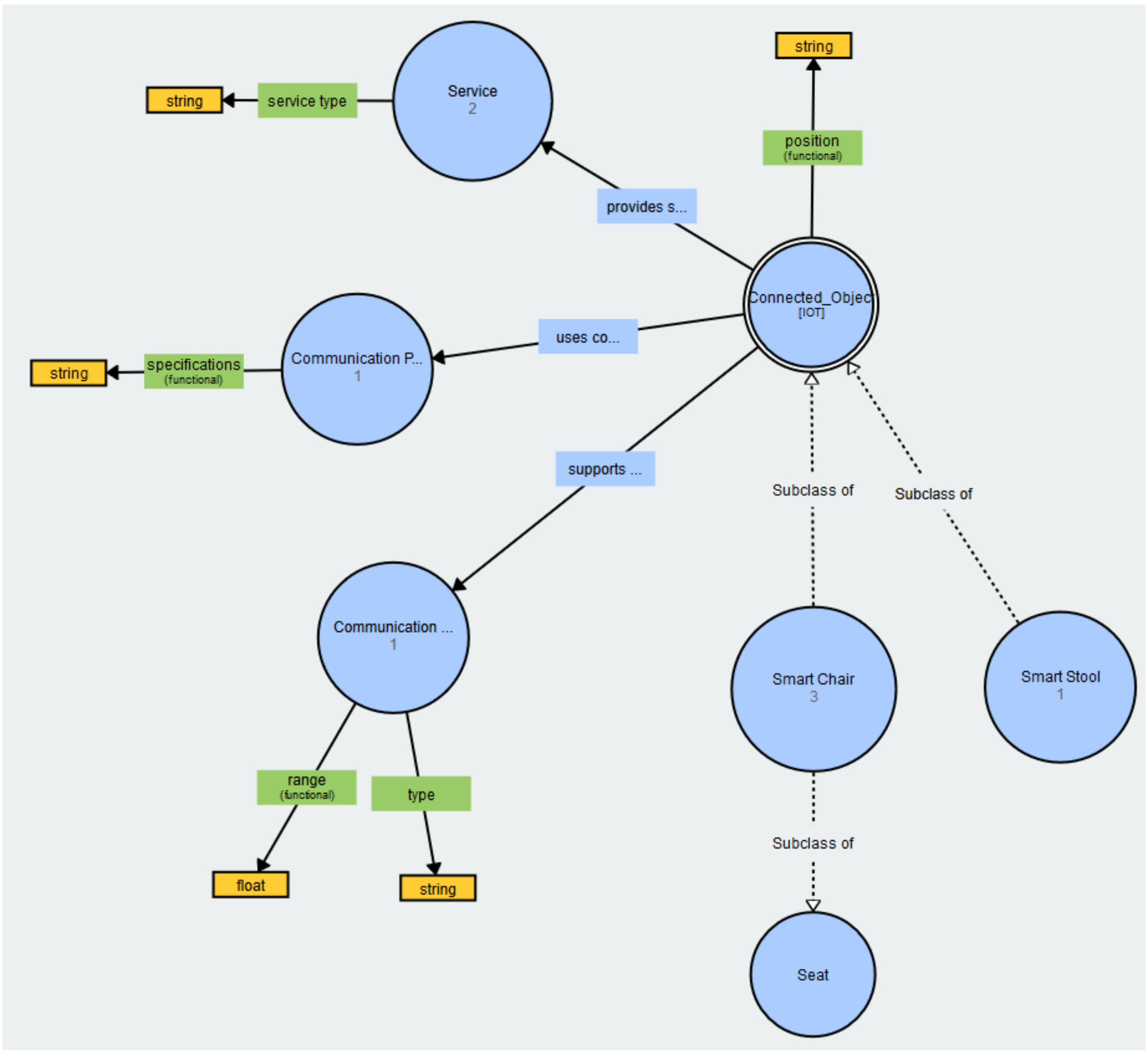
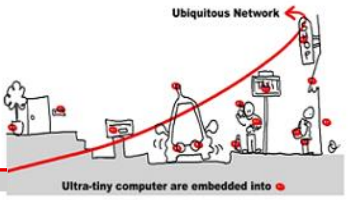


Figure 5 : The stool semantic annotations are merged with the knowledge base content. The Stool annotations define new concepts ('Smart Stool' and 'IOT') that were not previously defined in the knowledge base. The alignment process defines equivalence between concepts `Connected_Object` and `IOT`.

Tutorial: SWoT (Semantic Web of Things)



SPARQL is an RDF query language, not an OWL query language so has no understanding of inferences driven by OWL language (Some SPARQL implementations do allow querying of OWL entailments). The DL query language provided by protégé (`Window/Tabs/DL Query`) allows to deal with OWL language and queries will pick-up inferences. See http://protegewiki.stanford.edu/wiki/DLQueryTab for more details. Before querying using DL query language one need to start a reasoner. Protégé comes with a pre-installed reasoner (HermiT) that has to be started (`Reasoner/Start reasoner`).

Exercise#14
Once the reasoner has been started, open the DL Query tab, enter the below query. Do not forget to select 'Instances' from to right panel (Query for). Now execute the query. What do you remark?
Solution

```
Connected_Object and position value "Kitchen" and 'provides service' value Buzzer
```

Exercise#15
Execute a DL query to get all instances of smart chair.

Exercise#16
Modify `Alignment.owl` in order to add an equivalence between "`Smart Chair`" and "`Smart Stool`". Execute a DL query to get all instances of smart chair. What do you remark now?

Exercice#17
Extend the ontology with the concept `Person` and a property defining its position. Create two instances of this concept (Bob and Alice) and set their position in Kitchen and Office respectively.

Exercise#18
Execute a DL query to get all instances of connected objects providing the service `IsOccupied` around Bob and Alice.

8 References

[1]. Haller, S. (2010). **The things in the internet of things**. Poster at the (IoT 2010). Tokyo, Japan.
[2]. Dohr, A., Modre-Opsrian, R., Drobics, M., Hayn, D., & Schreier, G. (2010, April). **The internet of things for ambient assisted living**. In Information Technology: New Generations (ITNG), 2010 Seventh International Conference on (pp. 804-809). IEEE.
[3]. Atzori, L., Iera, A., & Morabito, G. (2010). **The internet of things: A survey**. Computer networks, 54(15), 2787 -2805.
[4]. Zeng, D., Guo, S., & Cheng, Z. (2011). **The web of things: A survey**. Journal of Communications, 6(6), 424-438.
[5]. Rocher, G., Tigli, J. Y., Lavirotte, S., & Daikhi, R. (2015, October). **Run-time knowledge model enrichment in SWoT: A step toward ambient services selection relevancy**. In Internet of Things (IOT), 2015 5th International Conference on the (pp. 62-69). IEEE.