

Éléments de correction du TP ontologie

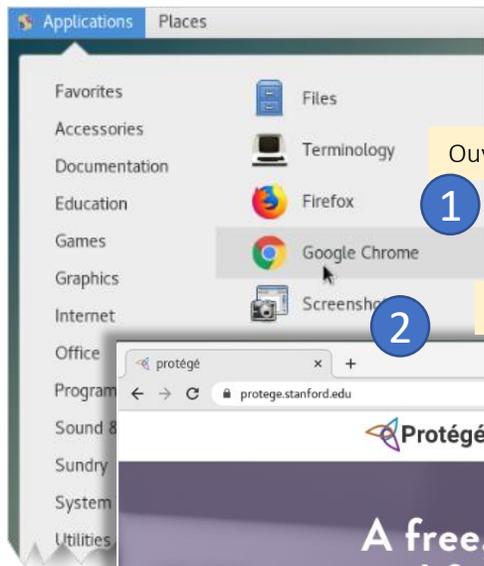
Pour un apprentissage de ces notions par l'exemple.

Installation de Protégé

<https://protege.stanford.edu/>

Installation de Protégé

- Double boot : choix windows ou linux
- Choisir de préférence windows l'installation est plus simple
- Prendre la version qui inclut java
- Onglet Documentation :
 - Protégé Desktop User Documentation
 - Editor features
 - Protégé Desktop Features

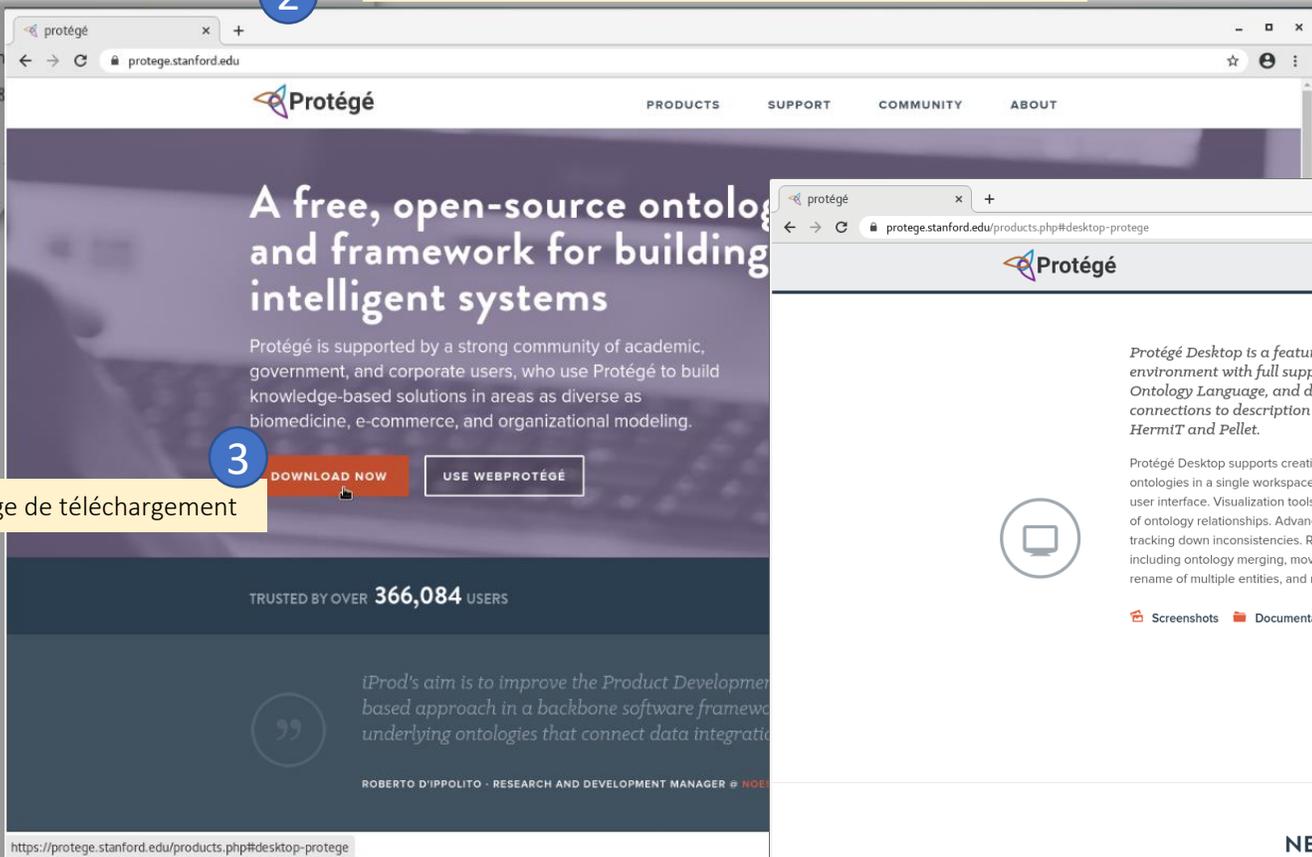


Ouvrir un navigateur (depuis menu Application – Internet)

1

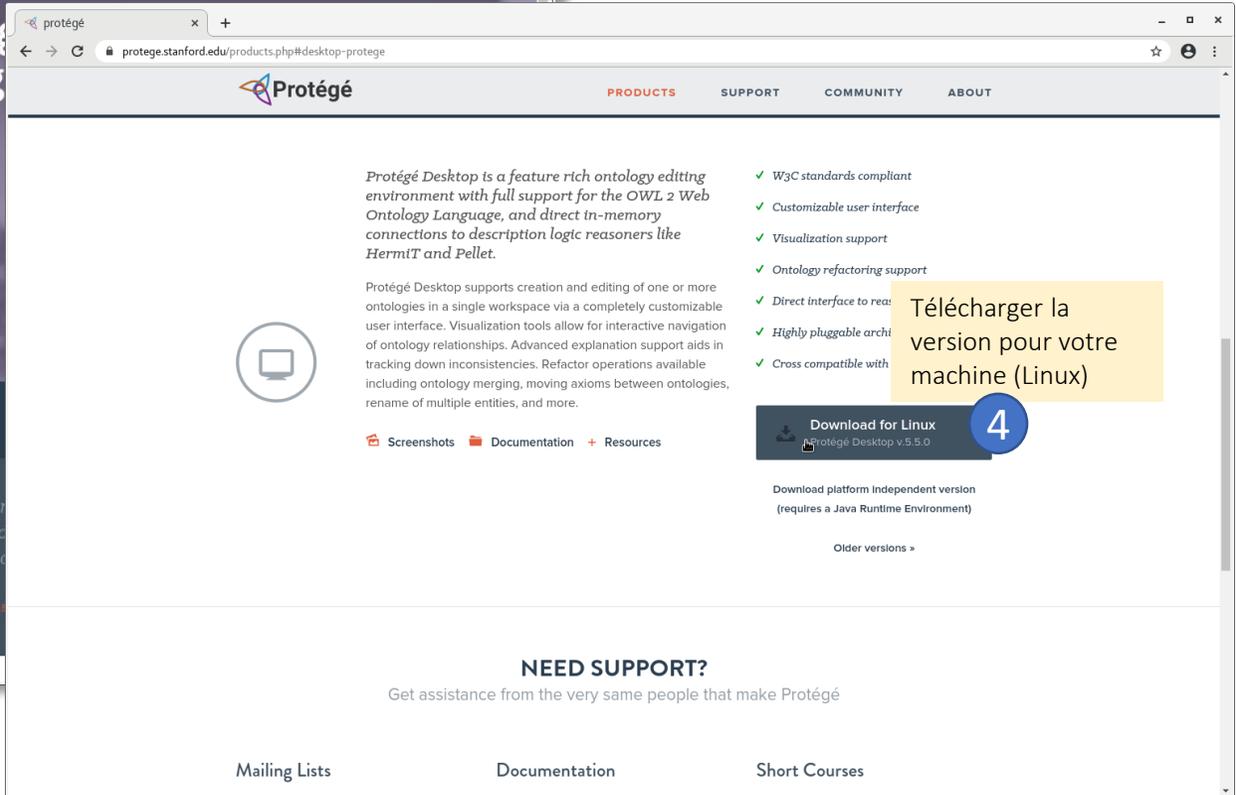
2

accéder à la page Protégé (protege.stanford.edu)



accéder à la page de téléchargement

3

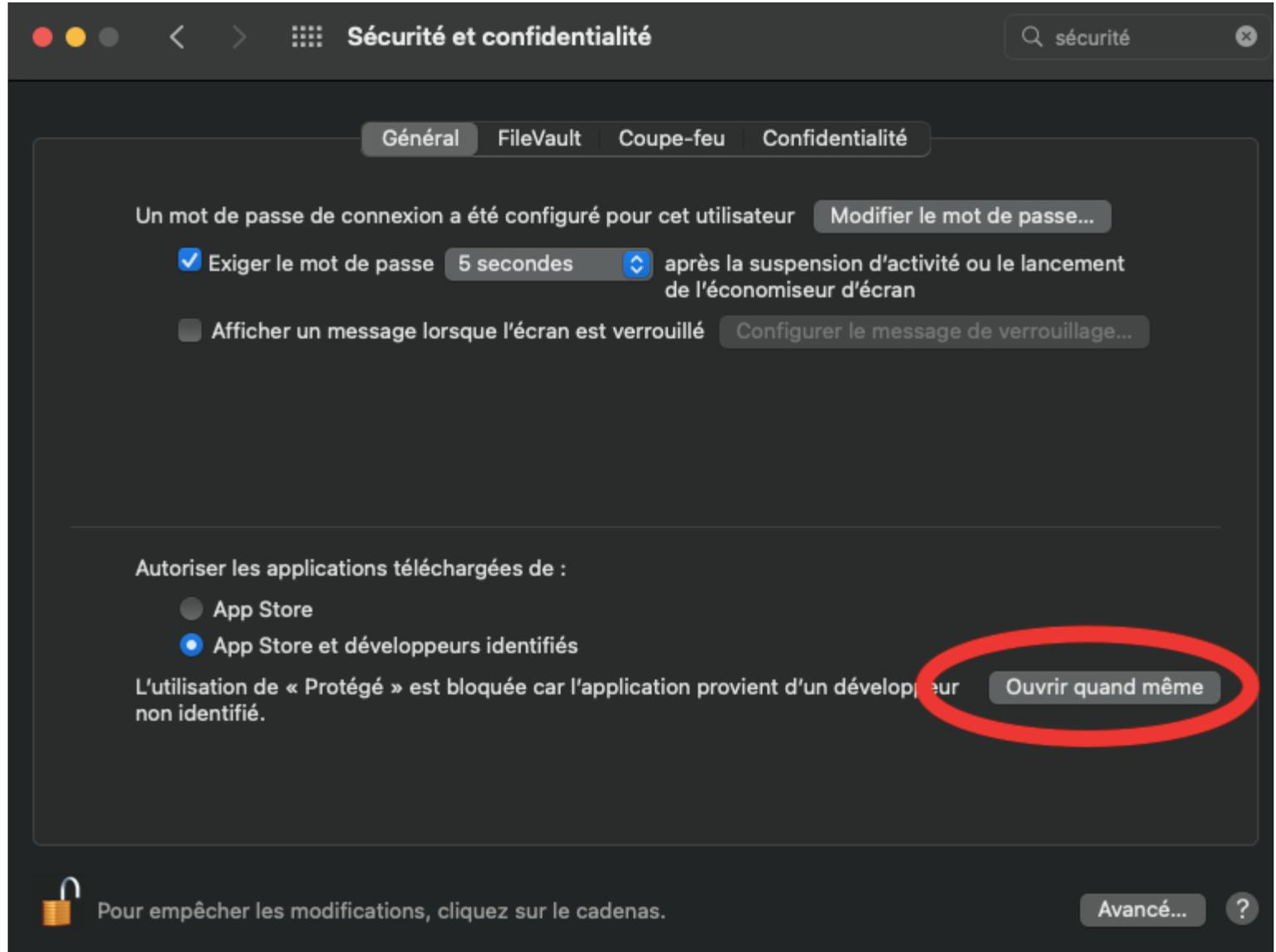
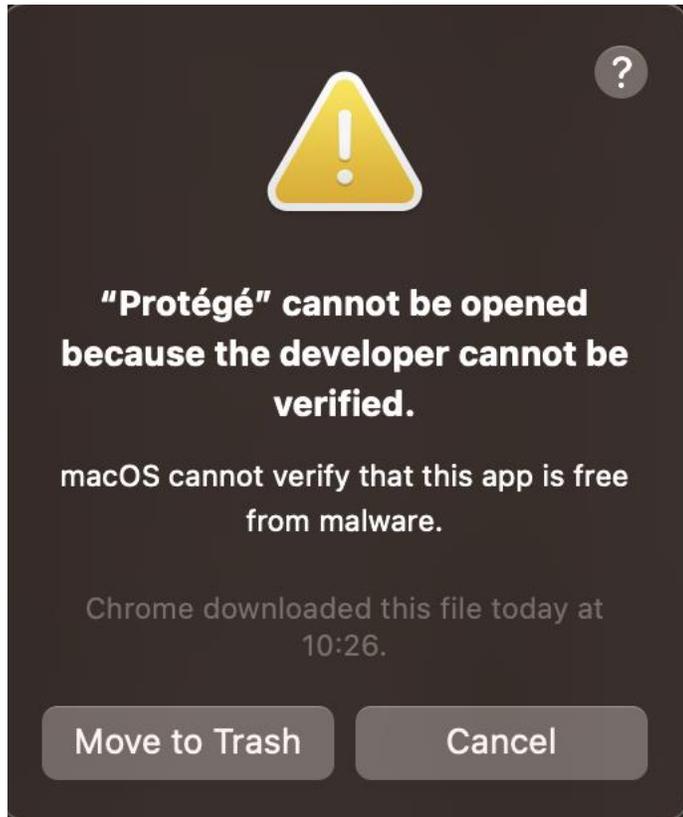


Télécharger la version pour votre machine (Linux)

4

Aide pour ceux qui choisissent Linux - Installation de Protégé sur les postes de travail -

Pour les macs



Décompressez l'archive puis ouvrez une fenêtre terminal

4

genoudph@ensipc435:/user/8/.base/genoudph/home/Protege-5.5.0

```
[genoudph@ensipc435 home]$ cd Protege-5.5.0/  
[genoudph@ensipc435 Protege-5.5.0]$ ls  
app bin bundles conf jre plugins run.sh  
[genoudph@ensipc435 Protege-5.5.0]$ ./run.sh
```

4

placez vous dans le répertoire de Protégé
cd Protege-5.5.0

5

pour lancer Protégé tapez la commande
./run.sh &

Aide pour ceux qui choisissent Linux - Installation de Protégé sur les postes de travail -

La fenêtre de l'application s'ouvre, vous pouvez commencer à travailler

6

The screenshot shows the Protégé application window titled "untitled-ontology-3". The interface includes a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help), a search bar, and several panels. The "Ontology header" panel shows the "Ontology IRI" as <http://www.semanticweb.org/genoudph/ontologies/2021/3/untitled-ontology-3> and the "Ontology Version IRI" as <http://www.semanticweb.org/genoudph/ontologies/2021/3/untitled-ontology-3>. The "Ontology metrics" panel displays a table of counts for various ontology elements:

Metrics	
Axiom	0
Logical axiom count	0
Declaration axioms count	0
Class count	0
Object property count	0
Data property count	0
Individual count	0
Annotation Property count	0
Class axioms	
SubClassOf	0
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
Object property axioms	

The "Imported ontologies" panel at the bottom shows "Direct Imports" and "Indirect Imports" sections, both currently empty. The status bar at the bottom indicates "No Reasoner set. Select a reasoner from the Reasoner menu" and "Show Inferences" is checked.

Charger le fichier IoT.owl

- Ouvrir le fichier avec Protégé
- Pour vous aider <http://protegeproject.github.io/protege/getting-started/>
- Ne pas double-cliquer sur l'URI
- <https://lig-membres.imag.fr/genoud/teaching/coursSW/IoT/IoT.owl>
<<https://lig-membres.imag.fr/genoud/teaching/coursSW/IoT/IoT.owl>>

Question 1.1 : Device

The screenshot displays a Semantic Web editor interface with three main panels:

- Class hierarchy: Device:** Shows a tree structure starting from `owl:Thing`. Under `owl:Thing` are `BlackBerry`, `Component`, `Device`, `Environnement`, and `Phone`. `Device` is highlighted.
- Description: Device:** Shows the class description for `Device`. It includes:
 - Equivalent To: (empty)
 - SubClass Of: `hasComponent some Component`
 - General class axioms: (empty)
 - SubClass Of (Anonymous Ancestor): (empty)
 - Instances: (empty)
 - Target for Key: (empty)
 - Disjoint With: `Component`
 - Disjoint Union Of: (empty)
- Usage: Device:** Shows 16 uses of the `Device` class. The uses are:
 - `Component`: `Component DisjointWith Device`
 - `Device`: `Class: Device`, `Component DisjointWith Device`, `Device SubClassOf hasComponent some Component`
 - `Environnement`: `Environnement SubClassOf Device`
 - `hasComponent`: `hasComponent Domain Device`
 - `isComponentOf`: `isComponentOf Range Device`
 - `Phone`: `Phone SubClassOf Device`



Classe annotée avec une contrainte

Classe avec des relations mais pas de contrainte sur ces relations

La classe Device est une sous-classe de la classe définie par les « Thing ». Elle a un composant, hasComponent (un certain Component). Elle est disjointe de "Component". Elle a deux sous-classes environnement, phone.

Classes disjointes

- Commentaires sur la notion de hasComponent ?
- Que signifie des classes disjointes ?

Question 1.2 : la propriété (*Object Propertie*) hasComponent

The screenshot displays a web-based ontology editor interface. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Individuals by class', and 'DL Query'. Below this, there are tabs for 'Data properties', 'Annotation properties', 'Datatypes', and 'Individuals'. The main content area is divided into three panels:

- Object property hierarchy: hasComponent:** Shows a tree view where 'hasComponent' is a sub-property of 'owl:topObjectProperty', and 'isComponentOf' is a sub-property of 'hasComponent'.
- Annotations: hasComponent:** A panel for adding annotations to the property.
- Description: hasComponent:** A panel for defining the property's characteristics and relationships. It includes a list of characteristics (Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, Irreflexive) and a list of relationships (Equivalent To, SubProperty Of, Inverse Of, Domains (intersection), Ranges (intersection), Disjoint With, SuperProperty Of (Chain)).

The 'Description: hasComponent' panel shows the following configuration:

- Equivalent To:** None
- SubProperty Of:** None
- Inverse Of:** isComponentOf
- Domains (intersection):** Device
- Ranges (intersection):** Component
- Disjoint With:** None
- SuperProperty Of (Chain):** None

- La propriété d'objet hasComponent : est une sous-propriété de la propriété owl :topObjectProperty, elle peut avoir comme domaine (Domain) les types d'entités " Device ", et comme image (Range) des entités de types « Component ». La propriété inverse de " hasComponent " est " isComponentOf ".

La hiérarchie des objectProperties

- hasComponent
 - hasEssentialComponent
 - hasQualityComponent
 - hasHightQualityComponent
 - ...

Question 1.3 : la propriété isComponentOf

The screenshot displays an ontology editor interface. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Individuals by class', and 'DL Query'. Below this, there are tabs for 'Data properties', 'Annotation properties', 'Datatypes', and 'Individuals'. The main content area is divided into several panes:

- Object property hierarchy: isComponentOf**: Shows a tree view where 'isComponentOf' is a sub-property of 'hasComponent', which is a sub-property of 'owl:topObjectProperty'.
- Annotations: isComponentOf**: A pane for adding annotations to the property.
- Characteristics**: A list of checkboxes for property characteristics: Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive.
- Description: isComponentOf**: A pane for defining the property's characteristics:
 - Equivalent To**: Empty.
 - SubProperty Of**: 'owl:topObjectProperty'.
 - Inverse Of**: 'hasComponent'.
 - Domains (intersection)**: 'Component'.
 - Ranges (intersection)**: 'Device'.
 - Disjoint With**: Empty.
 - SuperProperty Of (Chain)**: Empty.

- La propriété d'objet `isComponentOf` : est une sous-propriété de la propriété `owl:topObjectProperty`, elle peut avoir comme domaine (Domain) les types d'entités « Component », et comme image (Range) " Device ". La propriété inverse de " isComponentOf " est " hasComponent " .

Question 1.4 : la classe Phone

The screenshot displays an ontology editor interface with the following components:

- Class hierarchy: Phone:** A tree view showing the hierarchy: owl:Thing (parent) -> BlackBerry (child) -> Component (parent) -> Device (parent) -> Environment (child) -> Phone (child).
- Description: Phone:** A list of logical axioms for the class Phone:
 - Equivalent To: hasComponent some gsm
 - SubClass Of: Device
 - General class axioms: none
 - SubClass Of (Anonymous Ancestor): hasComponent some Component
 - Instances: none
 - Target for Key: none
 - Disjoint With: Component, Environment
 - Disjoint Union Of: none
- Usage: Phone:** A list of 18 uses of the class Phone:
 - Component: Component DisjointWith Phone
 - Environment: Environment DisjointWith Phone
 - Phone: Phone SubClassOf Device, Component DisjointWith Phone, Class: Phone, Environment DisjointWith Phone, Phone EquivalentTo hasComponent some gsm
 - Smartphone: Smartphone SubClassOf Phone
 - Telephone: Telephone SubClassOf Phone

La classe Phone :
est une sous-classe
de la classe
« Device », qui,
elle-même, est une
sous-classe de la
classe définie
« Thing ».
Elle hérite de la
propriété
hasComponent en
tant que sous classe
de Device. Elle est
disjointe des classes
« Composant » en
tant que sous-classe
de Device et de
« Environnement
» (en propre) et
hasComponent (s
ome gsm).

Question 1.5 : la classe SmartPhone

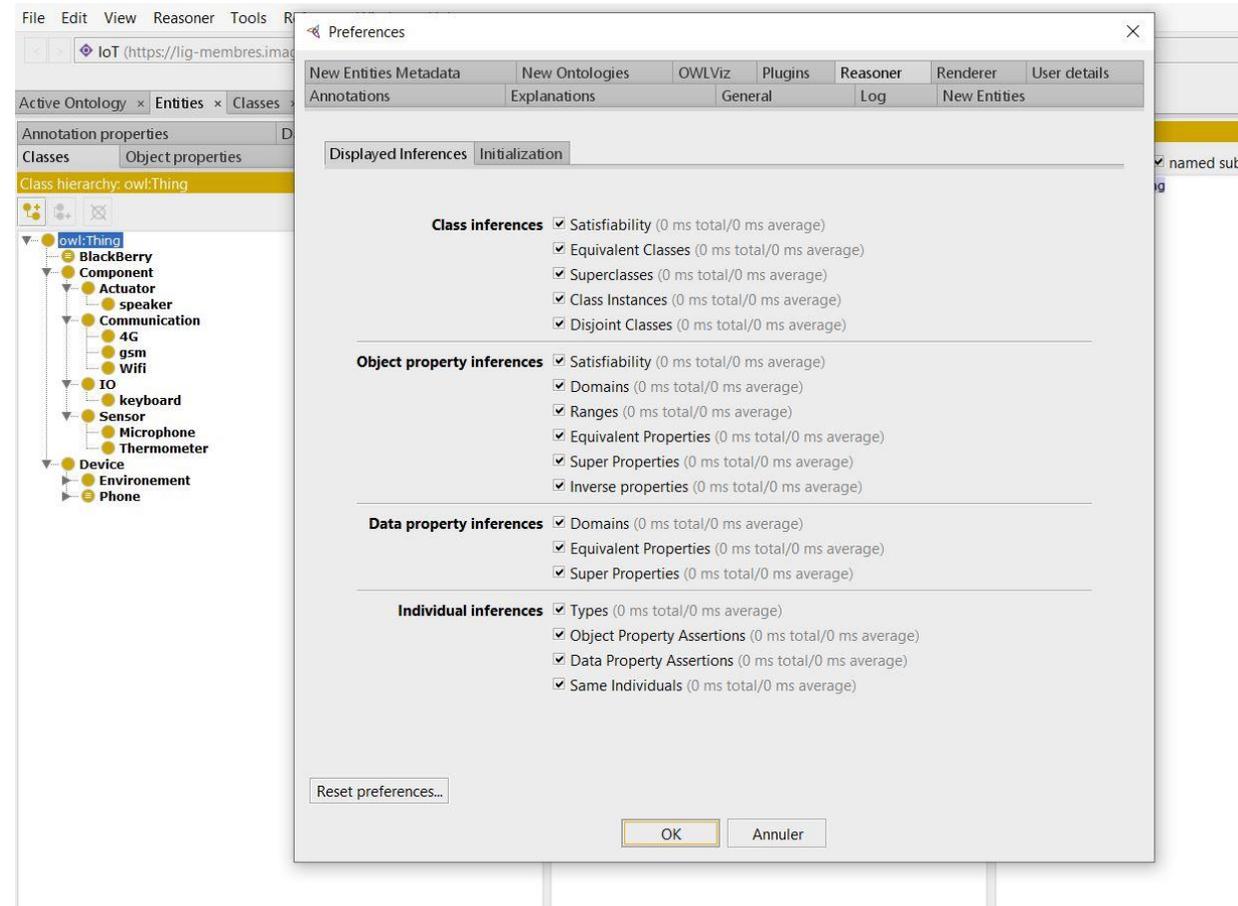
The screenshot displays a Semantic Web editor interface with three main panels:

- Class hierarchy: Smartphone:** A tree view showing the ontology structure. The hierarchy is: owl:Thing → Component → Actuator (with sub-classes: speaker) → Communication (with sub-classes: 4G, gsm, Wifi) → IO (with sub-classes: keyboard) → Sensor (with sub-classes: Microphone, Thermometer) → Device → Environment (with sub-classes: AirQualityStation, Hygrometer) → Phone (with sub-classes: Smartphone, Telephone).
- Description: Smartphone:** Shows logical axioms for the class:
 - Equivalent To: $(\text{hasComponent } \text{min } 1 \text{ Wifi}) \text{ and } (\text{hasComponent } \text{min } 1 \text{ gsm})$
 - SubClass Of: Phone
 - General class axioms: none listed
 - SubClass Of (Anonymous Ancestor): $\text{hasComponent } \text{some } \text{gsm}$ and $\text{hasComponent } \text{some } \text{Component}$
- Usage: Smartphone:** Shows the class's usage in the ontology:
 - Smartphone SubClassOf Phone
 - Class: Smartphone
 - Smartphone EquivalentTo $(\text{hasComponent } \text{min } 1 \text{ Wifi}) \text{ and } (\text{hasComponent } \text{min } 1 \text{ gsm})$

La classe SmartPhone est une sous-classe de "Phone". C'est aussi une sous-classe des deux classes définies par «Device" qui a un component (hasComponent) et (some gsm) et (2) qui ont un hasComponent (some Component) et some gsm par la classe phone. Elle est également équivalente à la classe définie par qui ont un «hasComponent» (min 1 Wifi) et hasComponent(min 1 gsm)".

Question 2: File > Préférences, onglet Reasoner, cochez toutes les cases et validez

Choisir **Hermit 1.4.3.456** et non ELK



Question 2 : Reasoner > Start reasoner

The screenshot displays an ontology editor interface with the following components:

- Top Tabs:** Active Ontology, Entities, Classes, Object Properties, Individuals by class, DL Query.
- Class Hierarchy Panel (Left):**
 - Class hierarchy: owl:Thing (Asserted)
 - Classes listed: BlackBerry, Component, Actuator (with sub-class speaker), Communication (with sub-classes 4G, gsm, Wifi), IO (with sub-class keyboard), Sensor (with sub-classes Microphone, Thermometer), Device, Environement (with sub-classes AirQualityStation, Hygrometer), Phone (with sub-classes Smartphone, Telephone).
- Annotations Panel (Top Right):** Annotations: owl:Thing. Contains a plus sign (+) for adding annotations.
- Description Panel (Bottom Right):** Description: owl:Thing. Contains several expandable sections:
 - Equivalent To (+)
 - SubClass Of (+)
 - General class axioms (+)
 - SubClass Of (Anonymous Ancestor)
 - Instances (+)
 - Target for Key (+)
 - Disjoint With (+)
 - Disjoint Union Of (+) - This section is highlighted in blue.

Question 2 : Raisonnement

The screenshot shows an ontology editor interface for the 'IoT' ontology. The left pane displays a class hierarchy where 'Smartphone' is a subclass of 'Phone', which is a subclass of 'Device'. The right pane shows the 'Description: Smartphone' section, which includes the following information:

- Equivalent To: $(\text{hasComponent } \text{min } 1 \text{ Wifi}) \text{ and } (\text{hasComponent } \text{min } 1 \text{ gsm})$
- SubClass Of: **Phone**
- General class axioms:
 - SubClass Of (Anonymous Ancestor): $\text{hasComponent } \text{some } \text{gsm}$ and $\text{hasComponent } \text{some } \text{Component}$
- Disjoint With: **Component** and **Environnement** (highlighted in yellow and circled in red)

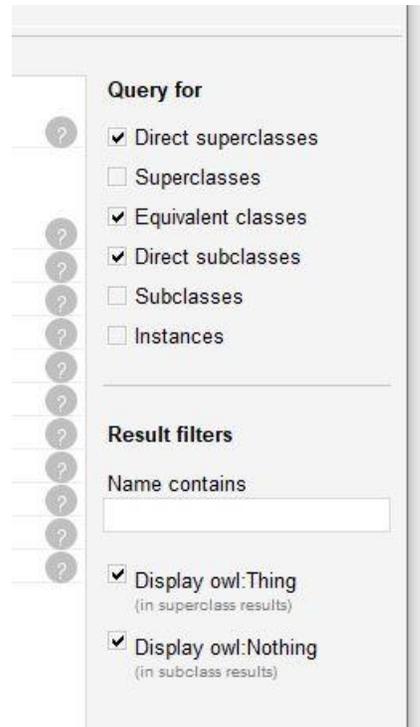
- Pour la classe Smartphone, de nouvelles connaissances sont ajoutées par inférence (en jaune dans l'éditeur) comme suit :
- - Disjoint avec Composant
- - Disjoint avec Environnement
- Ceci est dû au fait que :
 - Smartphone est une sous-classe de Phone.
 - Phone est disjoint de Component, et Phone est disjoint de Environnement.
 - Smartphone hérite donc des caractéristiques de sa superclasse, (à savoir Smartphone disjoint avec Component et Smartphone disjoint avec Environnement.

Question 2 : Raisonnement

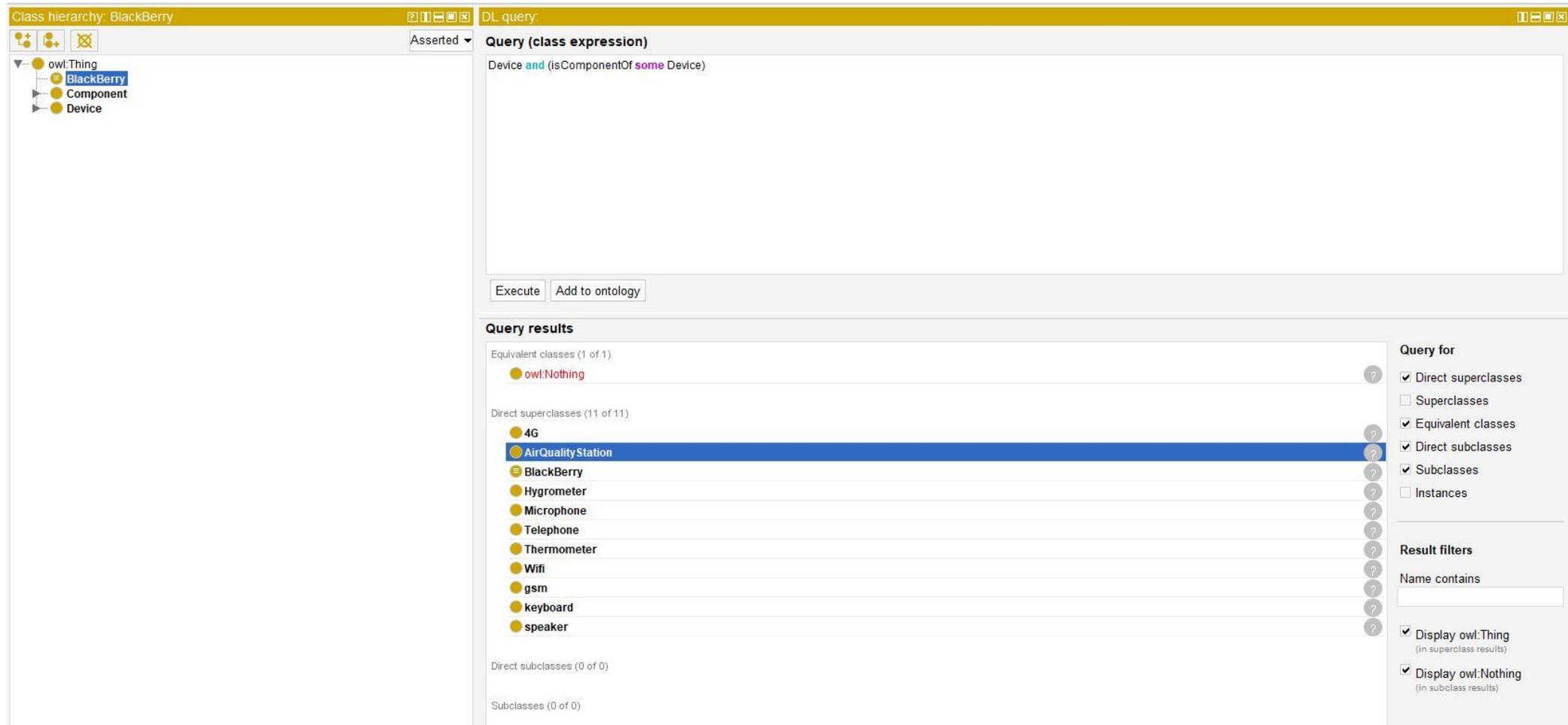
- Ensembles des classes modifiées :
 - Toutes les sous-classes de `Component` sont disjointes de `Device`
 - Il en est de même pour toutes les sous-classes de `Environnement` qui sont disjointes avec `Device`
 - `Environnement` étant disjoint avec `Phone`, ces sous-classes le sont aussi
 - Il en est de même pour toutes les sous-classes de `Phone` qui sont disjointes avec `Environnement`
 - `BlackBerry` est reconnu en tant que sous classe de `Smartphone` après inférence, du fait de sa description avant inférence
- Remarque : lorsqu'apparaissent des inconsistances : **owl:Nothing** en rouge est mentionné dans la fenêtre de navigation.

Question 3.0 : Query DL

- Window > Tabs, cochez DL Query,
- DL pour description logic
- Direct superclasses, Equivalent classes **et** Direct subclasses **sont bien cochées**



Question 3.0 :



Class hierarchy: BlackBerry

- owl:Thing
 - BlackBerry
 - Component
 - Device

DL query:

Query (class expression)
Device and (isComponentOf some Device)

Execute Add to ontology

Query results

Equivalent classes (1 of 1)

- owl:Nothing

Direct superclasses (11 of 11)

- 4G
- AirQualityStation
- BlackBerry
- Hygrometer
- Microphone
- Telephone
- Thermometer
- Wifi
- gsm
- keyboard
- speaker

Direct subclasses (0 of 0)

Subclasses (0 of 0)

Query for

- Direct superclasses
- Superclasses
- Equivalent classes
- Direct subclasses
- Subclasses
- Instances

Result filters

Name contains

- Display owl:Thing (in superclass results)
- Display owl:Nothing (in subclass results)

Pas de solution

Question 3.0 :

- Est-il possible d'avoir un Telephone avec une connexion Wifi ?
- En syntaxe de Manchester :
- Telephone and (hasComponent some wifi)

Q3.1 : Est-il possible d'avoir un Telephone avec une connexion Wifi ?

The screenshot shows the Protege software interface. On the left, a class hierarchy tree is visible with 'Telephone' selected. The main workspace contains a 'DL query' panel with the query: 'Telephone and hasComponent some Wifi'. Below the query, the 'Query results' section shows 'Subclasses (1 of 1)' and 'owl:Nothing'. On the right, there are 'Query for' and 'Result filters' options.

File Edit View Reasoner Tools Refactor Window Help

IoT (https://lig-membres.imag.fr/portet/IoT) : [F:\Enseignement\ENSIMAG-PHELMA-2A-ESI-SEOC\2023Cours-TD-TP-KR-2022-2023\IoT.owl]

Device > Phone > Telephone

Active ontology x Entities x Classes x Object properties x Data properties x Annotation properties x Individuals by class x DL Query x SWRLTab x

Class hierarchy: Telephone

- owl:Thing
 - BlackBerry
 - Component
 - Device
 - Environment
 - Phone
 - Smartphone
 - Telephone

DL query

Query (class expression)

Telephone and hasComponent some Wifi

Execute Add to ontology

Query results

Subclasses (1 of 1)

- owl:Nothing

Query for

- Direct superclasses
- Superclasses
- Equivalent classes
- Direct subclasses
- Subclasses
- Instances

Result filters

Name contains

Display owl:Thing (in superclass results)

Display owl:Nothing (in subclass results)

Pas de solution

Q3.2 : Est-il possible d'avoir un Smartphone sans Wifi ?

The screenshot shows a DL Query tool interface with the following components:

- Class hierarchy (Smartphone):** A tree view on the left showing the ontology structure. The **Smartphone** class is highlighted. Its hierarchy includes: **owl:Thing** (parent), **BlackBerry**, **Component**, **Actuator** (with **speaker** as a child), **Communication** (with **4G**, **gsm**, and **Wifi** as children), **IO** (with **keyboard** as a child), **Sensor** (with **Microphone** and **Thermometer** as children), **Device**, **Environment** (with **AirQualityStation** and **Hygrometer** as children), **Phone** (with **Smartphone** and **Telephone** as children).
- DL query:** A text area containing the query: `Smartphone and not (hasComponent some Wifi)`. Below the text area are buttons for **Execute** and **Add to ontology**.
- Query results:** A section showing the results of the query. It includes:
 - Equivalent classes (1 of 1):** **owl:Nothing**.
 - Direct superclasses (11 of 11):** A list of classes including **4G**, **AirQualityStation**, **BlackBerry**, **Hygrometer**, **Microphone**, **Telephone**, **Thermometer**, **Wifi**, **gsm**, **keyboard**, and **speaker**.
 - Direct subclasses (0 of 0):** A blue bar indicating no direct subclasses.
- Query for:** A panel on the right with checkboxes for **Direct superclasses** (checked), **Superclasses** (unchecked), **Equivalent classes** (checked), **Direct subclasses** (checked), **Subclasses** (unchecked), and **Instances** (unchecked).
- Result filters:** A section with a **Name contains** input field and checkboxes for **Display owl:Thing** (checked) and **Display owl:Nothing** (checked).

Pas de solution

Q3.3 : Un Smartphone a-t-il toujours un Microphone ?

The screenshot displays a software interface with two main panels. The left panel, titled "Class hierarchy: Smartphone", shows a tree structure of classes. The root is "owl:Thing", which branches into "BlackBerry", "Component", and "Device". "Component" includes "Actuator" (with "speaker" as a sub-class), "Communication" (with "4G", "gsm", and "Wifi" as sub-classes), and "IO" (with "keyboard" as a sub-class). "Device" includes "Sensor" (with "Microphone" and "Thermometer" as sub-classes), "Environment" (with "AirQuality Station" and "Hygrometer" as sub-classes), and "Phone" (with "Smartphone" and "Telephone" as sub-classes). The "Smartphone" class is highlighted in blue. The right panel, titled "DL query:", contains a text area with the query "Smartphone and not (hasComponent min 1 Microphone)". Below the text area are "Execute" and "Add to ontology" buttons. The "Query results" section shows three categories: "Equivalent classes (0 of 0)", "Direct superclasses (1 of 1)" with "Smartphone" listed, and "Direct subclasses (1 of 1)" with "owl:Nothing" listed. A "Query for" sidebar on the right has checkboxes for "Direct superclasses", "Superclasses", "Equivalent classes", "Direct subclasses", "Subclasses", and "Instances", with "Direct superclasses", "Equivalent classes", and "Direct subclasses" checked.

Pas de solution

Question 4 : Créer des individus

- On a besoin de `Individuals` du panneau `Description` pour rentrer les classes et `Property` assertions pour décrire les propriétés.
- `Window > Views > Individual views >` cocher les panneaux ci-dessus

4.1 Ajout : fairphone_1

The screenshot displays an ontology editor interface. At the top, there are tabs for 'Active ontology', 'Entities', 'Classes', 'Object properties', 'Annotation properties', 'Individuals by class', and 'DL Query'. Below these, there are sub-tabs for 'Annotation properties', 'Datatypes', 'Individuals', 'Classes', 'Object properties', and 'Data properties'. The main area shows a list of individuals under the heading 'Individuals: fairphone_1'. A single individual, 'fairphone_1', is listed and highlighted in blue. To the right, there are two panels. The top panel, titled 'Description: fairphone_1', shows a 'Types' section with a green plus sign and a yellow circle next to the class 'Device'. Below this is a 'Same Individual As' section with a plus sign. The bottom panel, titled 'Property assertions: fairphone_1', contains four sections: 'Object property assertions', 'Data property assertions', 'Negative object property assertions', and 'Negative data property assertions', each with a plus sign.

Idem pour micro_1 instance de microphone et gsm_1 instance de GSM

4.2 : “fairphone_1” ait “micro_1” comme composant ,
“gsm_1” soit un composant de “fairphone_1”

The screenshot displays the Protege ontology editor interface, showing the configuration of two individuals: 'fairphone_1' and 'micro_1'.

Left Panel (Individuals List): Shows a list of individuals under the 'fairphone_1' class: 'fairphone_1', 'gsm_1', and 'micro_1'.

Center Panel (Description: fairphone_1): Shows the configuration for the 'fairphone_1' individual. It is classified as a 'Device'. Under 'Property assertions', the 'hasComponent' property is asserted with the value 'gsm_1'.

Right Panel (Description: micro_1): Shows the configuration for the 'micro_1' individual. It is classified as a 'Microphone'. Under 'Property assertions', the 'isComponentOf' property is asserted with the value 'fairphone_1'.

Micro_1 isComponentOf fairphone_1
fairphone_1 hasComponent gsm_1

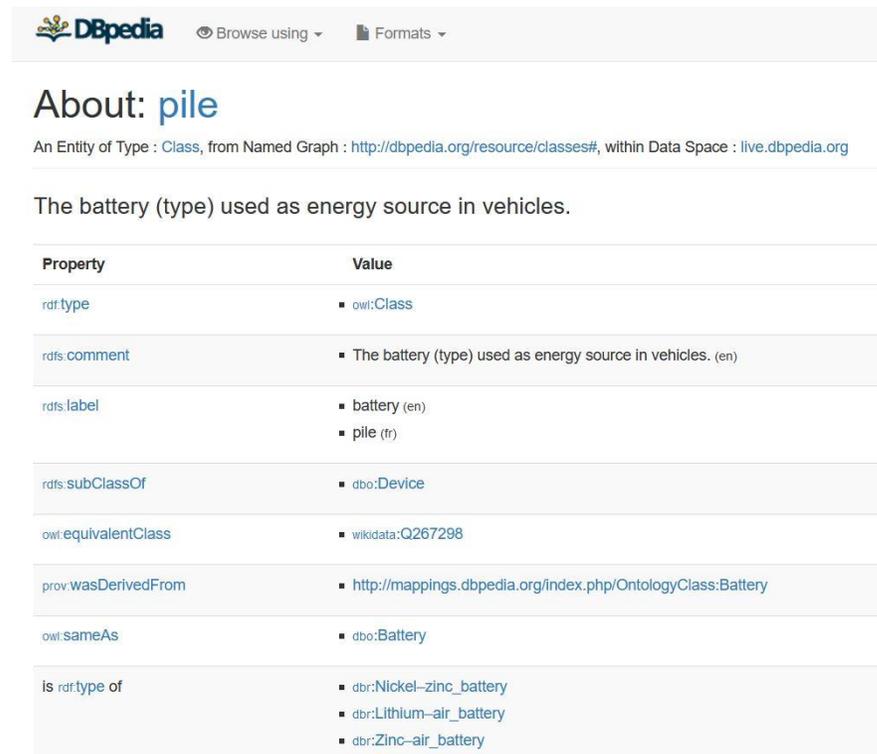
4.3 Inférence : Fairphone_1 isA Phone

The screenshot displays a Semantic Web browser interface with the following components:

- Navigation tabs:** Active ontology, Entities, Classes, Object properties, Annotation properties, Individuals by class, DL Query.
- Sub-navigation tabs:** Annotation properties, Datatypes, Individuals, Classes, Object properties, Data properties.
- Individuals panel:** Shows a list of individuals: fairphone_1 (selected), gsm_1, and micro_1.
- Description: fairphone_1:** A purple header with a search icon, a list icon, a refresh icon, and a close icon. Below it, a 'Types' section shows a list of classes: Device and Phone. The 'Phone' class is highlighted in yellow and circled in red. To the right of each class are icons for search, add, delete, and refresh.
- Property assertions: fairphone_1:** A purple header with a search icon, a list icon, a refresh icon, and a close icon. Below it, an 'Object property assertions' section shows two assertions: 'hasComponent gsm_1' and 'hasComponent micro_1'. The 'hasComponent micro_1' assertion is highlighted in yellow. To the right of each assertion are icons for search, add, delete, and refresh.
- Other sections:** 'Same Individual As', 'Data property assertions', 'Negative object property assertions', and 'Negative data property assertions', each with a plus icon to expand.

5.1 : web des données

- <https://dbpedia.org/ontology/Battery>



DBpedia Browse using Formats

About: pile

An Entity of Type : Class, from Named Graph : <http://dbpedia.org/resource/classes#>, within Data Space : live.dbpedia.org

The battery (type) used as energy source in vehicles.

Property	Value
<code>rdfs:type</code>	▪ <code>owl:Class</code>
<code>rdfs:comment</code>	▪ The battery (type) used as energy source in vehicles. (en)
<code>rdfs:label</code>	▪ <code>battery</code> (en) ▪ <code>pile</code> (fr)
<code>rdfs:subClassOf</code>	▪ <code>dbo:Device</code>
<code>owl:equivalentClass</code>	▪ <code>wikidata:Q267298</code>
<code>prov:wasDerivedFrom</code>	▪ http://mappings.dbpedia.org/index.php/OntologyClass:Battery
<code>owl:sameAs</code>	▪ <code>dbo:Battery</code>
<code>is rdfs:type of</code>	▪ <code>dbr:Nickel-zinc_battery</code> ▪ <code>dbr:Lithium-air_battery</code> ▪ <code>dbr:Zinc-air_battery</code>

5.2 Enrichir la base de connaissances

- <https://dbpedia.org/sparql>
- ou bien
- <http://live.dbpedia.org/sparql/>

The screenshot shows the Virtuoso SPARQL Query Editor interface. At the top, the browser address bar displays `live.dbpedia.org/sparql/`. Below the browser, there is a navigation bar with several icons and labels: "Les plus visités", "GraphDB Workbench", "Sign in · GitLab", "VocBench", and "dblp: computer scienc". The main title of the interface is "Virtuoso SPARQL Query Editor".

Below the title, there is a section for "Default Data Set Name (Graph IRI)" with a text input field containing `http://dbpedia.org`.

The "Query Text" section contains a text area with the following SPARQL query:

```
select distinct ?Concept where {[] a ?Concept} LIMIT 100
```

Below the query text, there is a section for "Results Format" with a dropdown menu set to "HTML".

The "Execution timeout" section has a text input field set to "30000" milliseconds, with a note: "milliseconds (values less than 1000 are ignored)".

The "Options" section contains several checkboxes:

- Strict checking of void variables
- Strict checking of variable names used in multiple clauses but not log
- Suppress errors on wrong geometries and errors on geometrical ope
- Log debug info at the end of output (has no effect on some queries a
- Generate SPARQL compilation report (instead of executing the quer

At the bottom of the interface, there is a note: "(The result can only be sent back to browser, not saved on the server, see [details](#))".

Finally, there are two buttons: "Run Query" and "Reset".

5.3 Enrichir la base de connaissances

- <https://dbpedia.org/sparql>
- ou bien
- <http://live.dbpedia.org/sparql/>

```
PREFIX dbo: <http://dbpedia.org/ontology/>  
CONSTRUCT {  
  ?subject ?predicate dbo:Battery  
}  
WHERE {  
  ?subject ?predicate dbo:Battery  
}
```

The screenshot shows the SPARQL Query Editor interface. At the top, there are navigation links: "SPARQL Query Editor", "About", "Tables", "Conductor", "Facet Browser", and "Permalink". Below this, the "Default Data Set Name (Graph IRI)" is set to "http://dbpedia.org". The "Query Text" area contains the SPARQL query: `PREFIX dbo: <http://dbpedia.org/ontology/>
CONSTRUCT {
 ?subject ?predicate dbo:Battery
}
WHERE {
 ?subject ?predicate dbo:Battery
}`. A yellow callout box with a red arrow points to the query text, labeled "Ecrivez votre requête SPARQL 1". Below the query text, the "Results Format" is set to "Turtle (beautified)", with a yellow callout box labeled "Choisissez le format de sortie Turtle 2". The "Execute Query" button is highlighted with a yellow callout box labeled "Exécutez la requête 3". A file dialog window titled "Ouverture de sparql_2021-10-14_20-49-55Z.ttl" is open, showing the file name and type. A yellow callout box labeled "Enregistrez le fichier .ttl (Turtle) sur votre machine 4" points to the "Enregistrer le fichier" option in the dialog. The dialog also shows "Ouvrir avec Notepad++" and "Toujours effectuer cette action pour ce type de fichier".

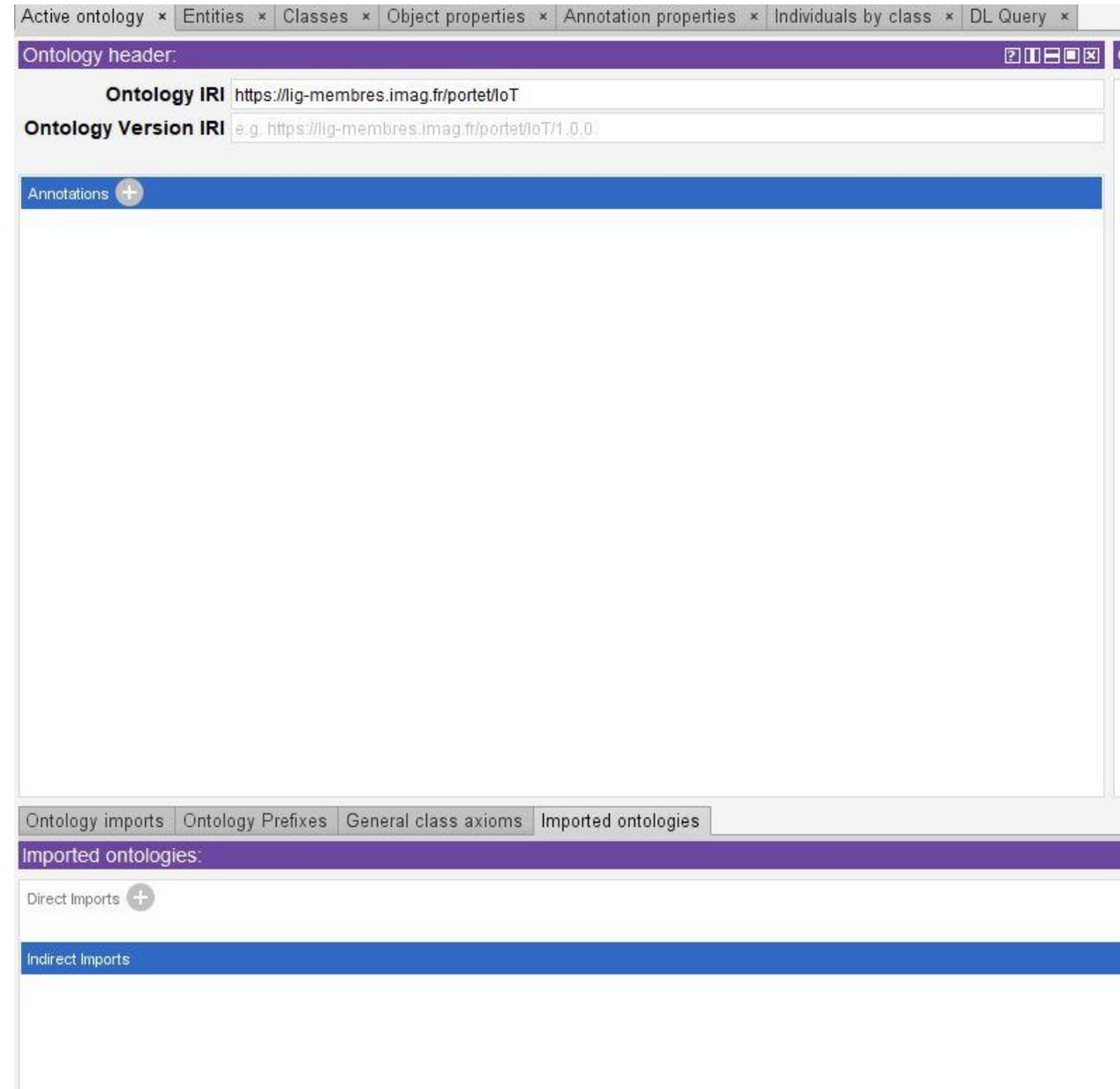
5.3c Enrichir la base de connaissances

- Enregistrer sous “resultat-1.ttl” sur la page de résultats.

```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dbo:      <http://dbpedia.org/ontology/> .
<http://dbpedia.org/resource/Nickel\u2013zinc_battery> rdf:type
    dbo:Battery .
<http://dbpedia.org/resource/Lithium\u2013air_battery> rdf:type
    dbo:Battery .
<http://dbpedia.org/resource/Zinc\u2013air_battery>      rdf:type
    dbo:Battery .
@prefix dbr:      <http://dbpedia.org/resource/> .
dbr:Silver-oxide_battery      rdf:type      dbo:Battery .
dbr:Superconducting_magnetic_energy_storage rdf:type
    dbo:Battery .
dbr:Lithium-ion_battery      rdf:type      dbo:Battery .
dbr:Alkaline_battery      rdf:type      dbo:Battery .
<http://dbpedia.org/resource/Nickel\u2013cadmium_battery>
    rdf:type      dbo:Battery .
<http://dbpedia.org/resource/Aluminium\u2013air_battery>
    rdf:type      dbo:Battery .
<http://dbpedia.org/resource/Nickel\u2013iron_battery> rdf:type
    dbo:Battery .
<http://dbpedia.org/resource/Nickel\u2013hydrogen_battery>
    rdf:type      dbo:Battery .
<http://dbpedia.org/resource/Nickel
\u2013metal_hydride_battery>      rdf:type      dbo:Battery .
dbr:Lithium-ion_capacitor      rdf:type      dbo:Battery .
<http://dbpedia.org/resource/Lithium\u2013titanate_battery>
    rdf:type      dbo:Battery .
<http://dbpedia.org/resource/Lead\u2013acid_battery>      rdf:type
    dbo:Battery .
dbr:Vanadium_redox_battery      rdf:type      dbo:Battery .
dbr:Lithium_iron_phosphate_battery      rdf:type      dbo:Battery .
<http://dbpedia.org/resource/Zinc\u2013bromine_battery>
    rdf:type      dbo:Battery .
<http://dbpedia.org/resource/Lithium\u2013sulfur_battery>
    rdf:type      dbo:Battery .
dbr:Lithium-titanate_battery      rdf:type      dbo:Battery .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
```

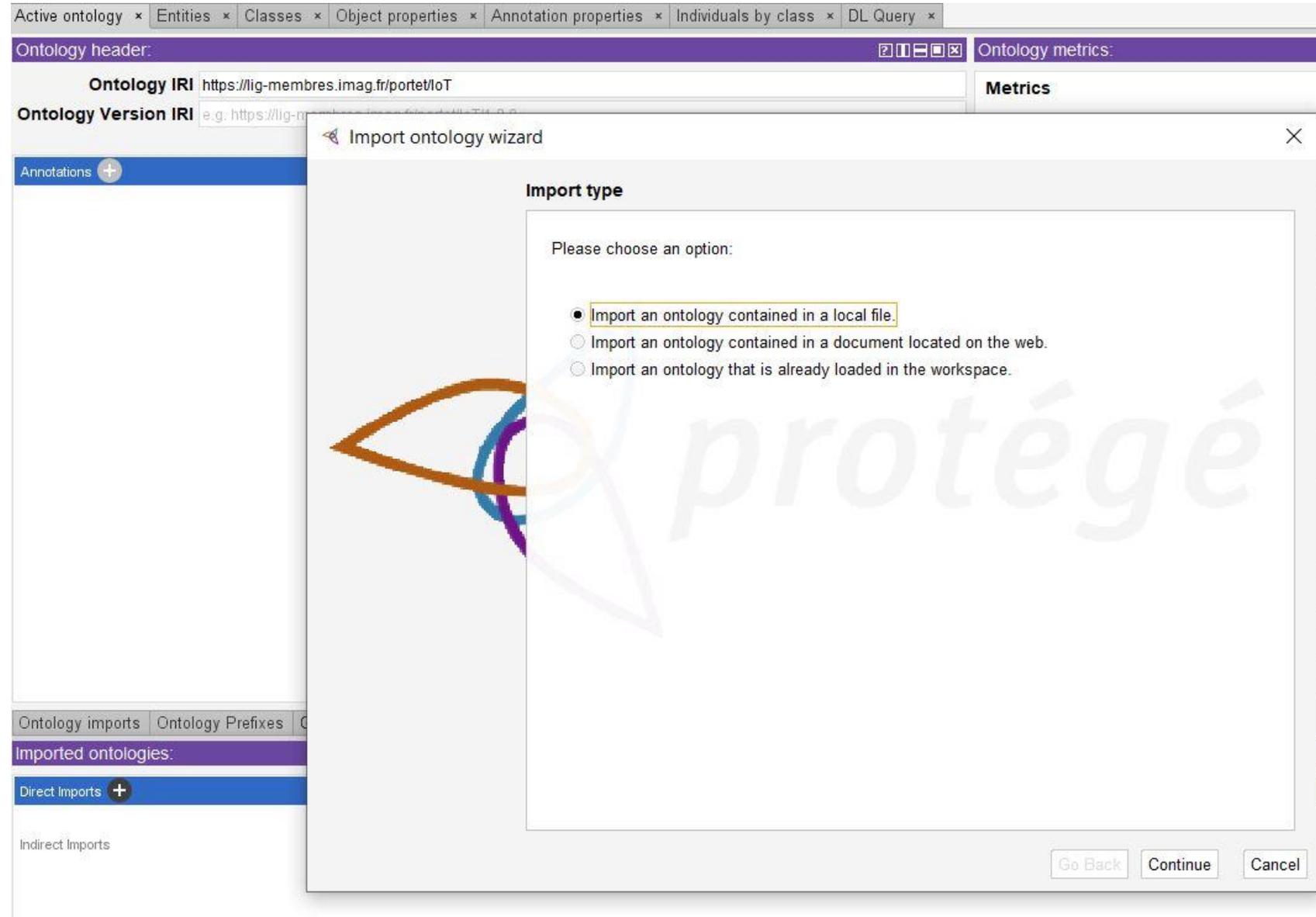
5.4a Incorporer ces éléments dans l'ontologie

- Arrêter le raisonneur : menu Reasoner > Stop reasoner
- Active Ontology,
- Panneau Window > Ontology view > Imported Ontology



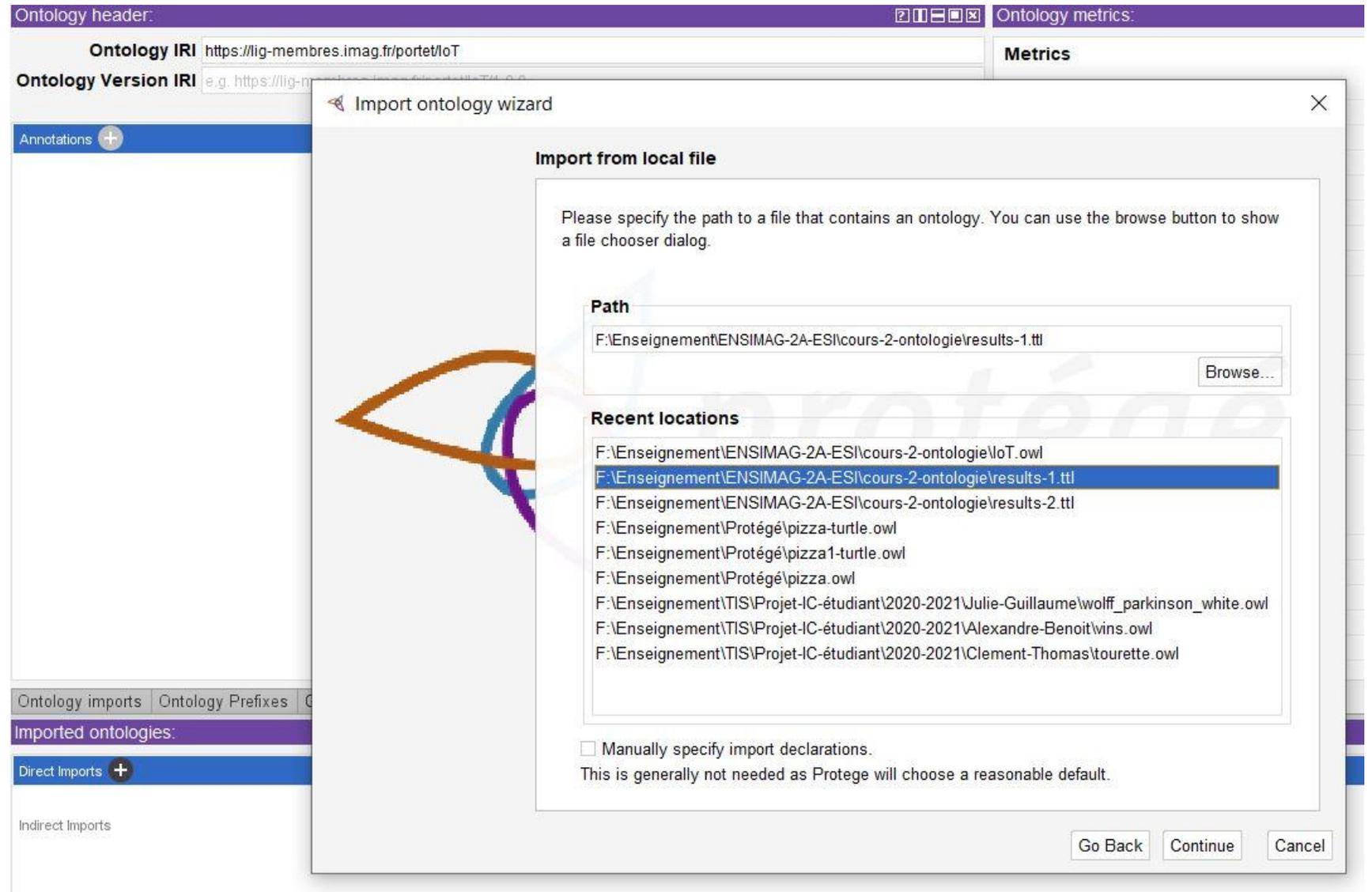
5.4b Incorporer ces éléments dans l'ontologie

- cliquez sur Direct Imports + -> Import an ontology contained in a local file -> Continue -> Browse



5.4c Incorporer ces éléments dans l'ontologie

- cliquez sur Direct Imports + -> Import an ontology contained in a local file -> Continue -> Browse



5.4d Incorporer ces éléments dans l'ontologie

The screenshot displays a web-based ontology editor interface. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Classes', 'Object properties', 'Annotation properties', 'Individuals by class', and 'DL Query'. Below this, there are sub-tabs for 'Annotation properties', 'Datatypes', 'Individuals', 'Classes', 'Object properties', and 'Data properties'. The main content area is divided into three panels:

- Class hierarchy: dbo:Battery:** A tree view showing the hierarchy of classes. The root is 'owl:Thing', which branches into 'BlackBerry', 'Component', 'Device', 'Environnement', and 'Phone'. 'Component' further branches into 'Actuator', 'Communication', 'IO', and 'Sensor'. 'Communication' includes '4G', 'gsm', and 'Wifi'. 'IO' includes 'keyboard'. 'Sensor' includes 'Microphone' and 'Thermometer'. 'dbo:Battery' is highlighted in blue.
- Description: dbo:Battery:** A panel for defining the class description. It includes sections for 'Types', 'Same Individual As', and 'Different Individuals', each with a plus sign for adding new entries.
- Property assertions:** A panel for defining property assertions. It includes sections for 'Object property assertions', 'Data property assertions', 'Negative object property assertions', and 'Negative data property assertions', each with a plus sign for adding new entries.

On the right side, there are two panels:

- Description: dbo:Battery:** A panel for defining the class description. It includes sections for 'Equivalent To', 'SubClass Of', 'General class axioms', and 'SubClass Of (Anonymous Ancestor)', each with a plus sign for adding new entries.
- Instances:** A list of instances of the class 'dbo:Battery'. Each instance is represented by a diamond icon, a label, and three small icons (a question mark, an '@' symbol, and an 'x'). The instances listed are: 'dbpedia:Alkaline_battery', 'dbpedia:Aluminium%E2%80%93air_battery', 'dbpedia:Aluminium-air_battery', 'dbpedia:Lead%E2%80%93acid_battery', 'dbpedia:Lead-acid_battery', 'dbpedia:Lithium%E2%80%93air_battery', 'dbpedia:Lithium-ion_battery', 'dbpedia:Lithium-ion_capacitor', and 'dbpedia:Lithium%E2%80%93sulfur_battery'.

At the bottom right, there is a panel for 'Usage: dbo:Battery' with a 'Show:' section containing checkboxes for 'this', 'disjoints', and 'named sub/superclasses'. Below this, it says 'Found 64 uses of dbo:Battery'.

5.5 Incorporer ces éléments dans l'ontologie

The screenshot displays a web-based ontology editor interface. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Classes', 'Object properties', 'Annotation properties', 'Individuals by class', and 'DL Query'. Below this, there are sub-tabs for 'Annotation properties', 'Datatypes', and 'Individuals', with further sub-tabs for 'Classes', 'Object properties', and 'Data properties'. The main content area is divided into three panels:

- Class hierarchy: dbo:Battery:** A tree view showing the ontology structure. The root is 'owl:Thing', which branches into 'BlackBerry', 'Component', 'Actuator', 'Communication', 'IO', 'Sensor', 'Device', 'Environnement', and 'Phone'. Under 'Actuator', there are 'dbo:Battery', 'speaker', and 'Communication'. Under 'Communication', there are '4G', 'gsm', and 'Wifi'. Under 'IO', there are 'keyboard' and 'Sensor'. Under 'Sensor', there are 'Microphone' and 'Thermometer'. Under 'Device', there are 'Environnement' and 'Phone'.
- Description: (purple header):** A panel for defining the class. It includes sections for 'Types', 'Same Individual As', and 'Different Individuals', each with a plus sign for adding new entries.
- Property assertions: (purple header):** A panel for defining property assertions. It includes sections for 'Object property assertions', 'Data property assertions', 'Negative object property assertions', and 'Negative data property assertions', each with a plus sign for adding new entries.

On the right side, there is a detailed view for the selected class 'dbo:Battery' (yellow header). This view includes:

- Description: dbo:Battery:** A panel showing the class's relationships. It includes sections for 'Equivalent To', 'SubClass Of' (with 'Actuator' listed), and 'General class axioms'.
- Instances:** A list of instances of the class, each with a plus sign for adding more instances and a minus sign for removing them. The instances listed are: 'dbpedia:Alkaline_battery', 'dbpedia:Aluminium%E2%80%93air_battery', 'dbpedia:Aluminium-air_battery', 'dbpedia:Lead%E2%80%93acid_battery', 'dbpedia:Lead-acid_battery', 'dbpedia:Lithium%E2%80%93air_battery', 'dbpedia:Lithium%E2%80%93sulfur_battery', and 'dbpedia:Lithium-ion_battery'.
- Usage: dbo:Battery:** A panel showing the usage of the class. It includes a 'Show:' section with checkboxes for 'this', 'disjoints', and 'named sub/superclasses'. Below this, it states 'Found 66 uses of dbo:Battery'.

5.6a Fairphone_1 hasComponent dbpedia:Lithium-ion_battery

- ARRETER LE RAISONNEUR

5.6b Fairphone_1 hasComponent dbpedia:Lithium-ion_battery

The screenshot displays a Semantic Web browser interface with the following components:

- Navigation tabs:** Active ontology, Entities, Classes, Object properties, Annotation properties, Individuals by class, DL Query.
- Sub-panels:** Classes, Object properties, Data properties, Annotation properties, Datatypes, Individuals.
- Individuals: fairphone_1:** A list of individuals including various battery types (e.g., Alkaline_battery, Lithium-ion_battery) and other devices (gsm_1, micro_1). The **fairphone_1** individual is highlighted in blue.
- Description: fairphone_1:** Shows the type **Device**.
- Property assertions: fairphone_1:** Lists object property assertions: **hasComponent dbpedia:Lithium-ion_battery** and **hasComponent gsm_1**.

5.7a Modifier la requête sur dbpedia

- <https://dbpedia.org/sparql>

ou bien

- <http://live.dbpedia.org/sparql/>

```
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
CONSTRUCT {
```

```
    ?subject ?predicate ?object
```

```
}
```

```
WHERE {
```

```
    dbo:Battery ?predicate ?object
```

```
}
```

5.7 Enrichir la base de connaissances

- Enregistrer sous “resultat-2.ttl” sur la page de résultats.

```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dbo:      <http://dbpedia.org/ontology/> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
dbo:Battery       rdf:type      owl:Class .
@prefix wikidata: <http://www.wikidata.org/entity/> .
dbo:Battery       owl:equivalentClass  wikidata:Q267298 .
@prefix prov:     <http://www.w3.org/ns/prov#> .
@prefix ns5:      <http://mappings.dbpedia.org/index.php/OntologyClass:> .
dbo:Battery       prov:wasDerivedFrom  ns5:Battery .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
dbo:Battery       rdfs:subClassOf  dbo:Device ;
                  rdfs:comment    "The battery (type) used as energy source
in vehicles."@en ;
                  rdfs:label      "bater\u00E9a"@es ,
                                "batterij"@nl ,
                                "bateria"@pt ,
                                "batteria"@it ,
                                "pile"@fr ,
                                "battery"@en ,
                                "Batterie"@de .
```

Avant 5.8 Stopper le raisonneur

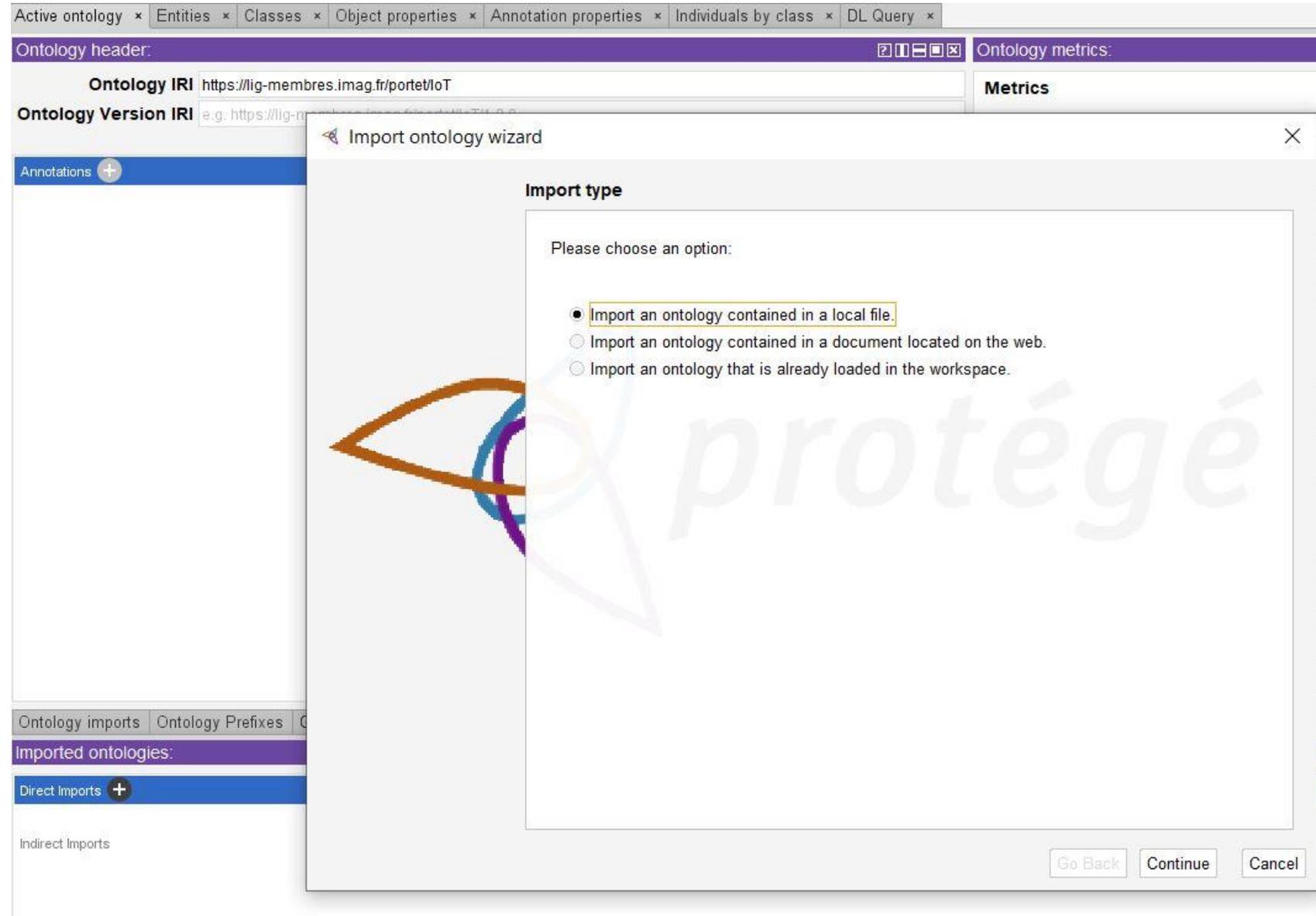
5.8a Incorporer ces éléments dans l'ontologie

- Active Ontology,
- Panneau Window >
Ontology view >
Imported Ontology,

The screenshot displays the Protégé ontology editor interface. At the top, there are several tabs: 'Active ontology', 'Entities', 'Classes', 'Object properties', 'Annotation properties', 'Individuals by class', and 'DL Query'. Below the tabs, the 'Ontology header' section is visible, containing two input fields: 'Ontology IRI' with the value 'https://lig-membres.imag.fr/portet/IoT' and 'Ontology Version IRI' with the value 'e.g. https://lig-membres.imag.fr/portet/IoT/1.0.0'. Below the header, there is a large empty area labeled 'Annotations' with a plus sign icon. At the bottom, there are several tabs: 'Ontology imports', 'Ontology Prefixes', 'General class axioms', and 'Imported ontologies'. The 'Imported ontologies' tab is currently selected, showing a section labeled 'Imported ontologies:' with two sub-sections: 'Direct Imports' (with a plus sign icon) and 'Indirect Imports'.

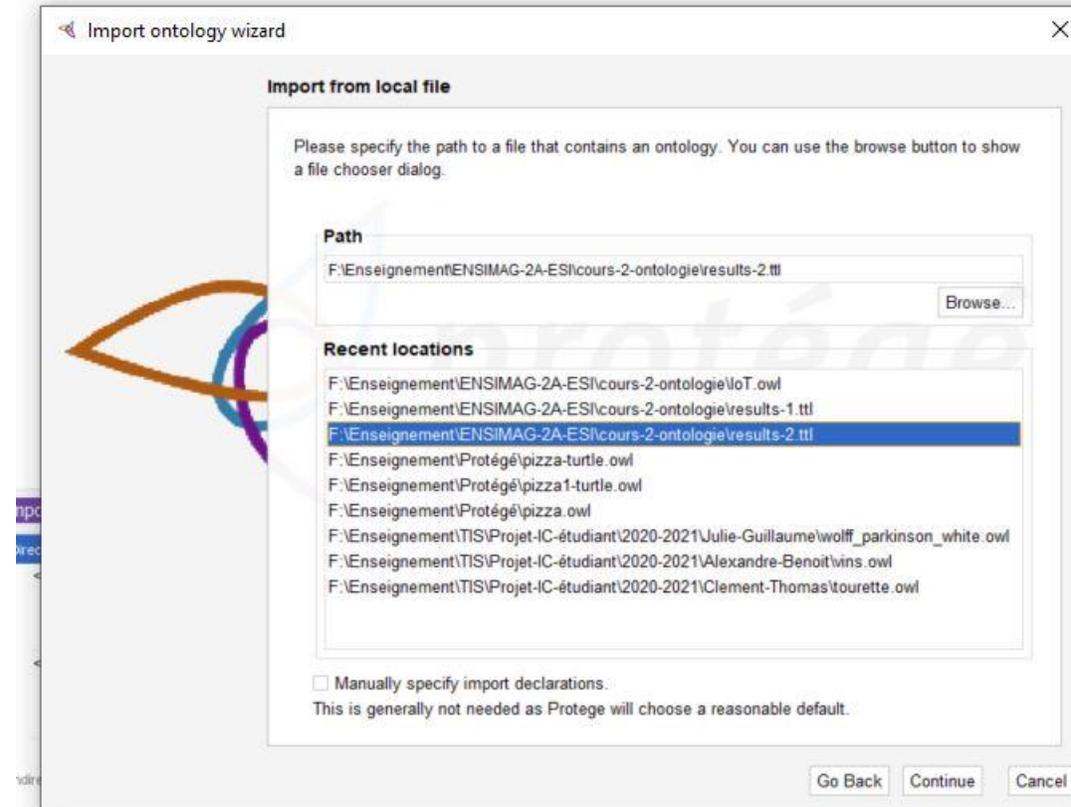
5.8b Incorporer ces éléments dans l'ontologie

- Fenetre imported ontologies
- cliquez sur Direct Imports + -> Import an ontology contained in a local file -> Continue -> Browse



5.8c Incorporer ces éléments dans l'ontologie

- cliquez sur Direct Imports + -> Import an ontology contained in a local file -> Continue -> Browse



5.9 Incorporer ces éléments dans l'ontologie

The screenshot displays the Protégé ontology editor interface. The top menu bar includes 'Active ontology', 'Entities', 'Classes', 'Object properties', 'Annotation properties', 'Individuals by class', and 'DL Query'. Below this, a secondary menu bar shows 'Classes', 'Object properties', 'Data properties', 'Annotation properties', 'Datatypes', and 'Individuals'. The main window is divided into three panes:

- Class hierarchy: pile**: A tree view showing the ontology structure. The 'pile' class is highlighted under the 'Actuator' class, which is a subclass of 'Component'. Other classes include 'BlackBerry', 'Communication' (with subclasses '4G', 'gsm', 'Wifi'), 'IO' (with subclass 'keyboard'), 'Sensor' (with subclasses 'Microphone', 'Thermometer'), 'dbo:Device', 'Device', 'Environment' (with subclasses 'AirQualityStation', 'Hygrometer'), and 'Phone' (with subclasses 'Smartphone', 'Telephone').
- Description: pile**: A panel showing the class description. It includes:
 - Equivalent To**: Q267298
 - SubClass Of**: Actuator, dbo:Device
 - General class axioms**: +
 - SubClass Of (Anonymous Ancestor)**: pile
 - Instances**: dbpedia:Alkaline_battery, dbpedia:Aluminium%E2%80%93air_battery, dbpedia:Aluminium-air_battery, dbpedia:Lead%E2%80%93acid_battery, dbpedia:Lead-acid_battery, dbpedia:Lithium%E2%80%93air_battery, dbpedia:Lithium-air_battery
- Usage: pile**: A panel showing the usage of the 'pile' class. It includes:
 - Show:** this disjoints named sub/superclasses
 - Found 101 uses of pile**
 - Usage examples:**
 - dbo:battery Range pile
 - dbpedia:Alkaline_battery Type pile
 - dbpedia:Aluminium%E2%80%93air_battery Type pile
 - dbpedia:Aluminium-air_battery Type pile
 - dbpedia:Lead%E2%80%93acid_battery Type pile
 - dbpedia:Lead-acid_battery Type pile
 - dbpedia:Lithium%E2%80%93air_battery Type pile
 - dbpedia:Lithium-air_battery Type pile

5.9 commentaires sur l'ontologie

- Incohérente car :
 - Pile est sous classe de Actuator est donc de Component
 - Pile est aussi sous-classe de dbo:device
 - dbo:device est équivalente à device

L'exécution du raisonneur aboutit à une ""ontologie incohérente"". Cela provient de :

D'un côté :

battery est une subclassOf dbo:Device (triplet importé)

et battery est aussi une subclassOf Actuator (nous l'avons classé)

et Actuator est une sous-classe de Component.

Donc Battery est une sous-classe de Component.

dbpedia:Lithium-ion_battery est un (Type) Battery, alors dbpedia:Lithium-ion_battery est aussi un (Type) Component.

D'un autre côté :

dbo:Device est équivalent à Device (nous l'avons inséré maintenant) ce qui signifie que battery est subclassOf Device (par substitution)

dbpedia:Lithium-ion_battery est un (Type) Batterie, alors dbpedia:Lithium-ion_battery est aussi un (Type) Dispositif

Maintenant, nous avons :

dbpedia:Lithium-ion_battery est aussi un (Type) Composant

dbpedia:Lithium-ion_battery est aussi un (Type) Dispositif

Mais Device et Component sont disjoints, alors, contradiction, alors une ontologie incohérente.

6 et 7. Ajout de classes

- Sous classe de speaker Smartspeaker
- Il convient d'ajouter la classe « «Vaccum » as a subclassOf «Device » ,
et ensuite d'ajouter « SmartVaccum » as a subclassOf « Vaccum ».