

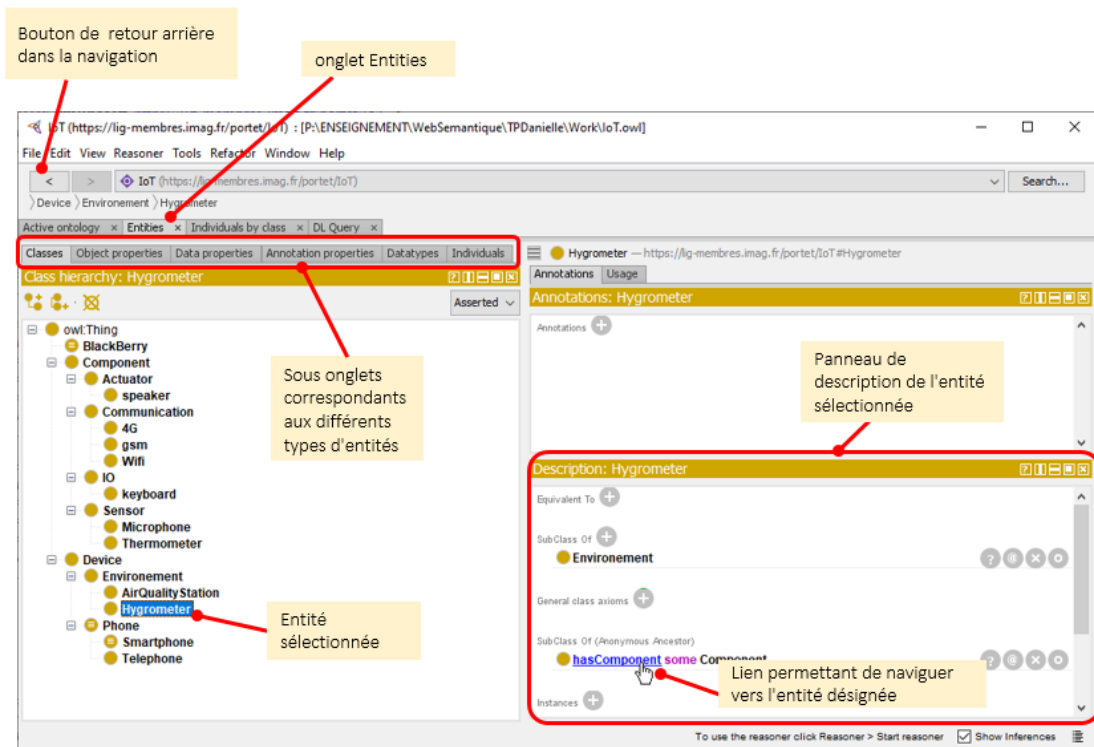
# TP Ontologie

(D'après un document de F. Portet)

## Préparation

1. Utilisez votre poste en windows de préférence
  2. Téléchargez l'ontologie `IoT.owl` que vous pouvez récupérer sur [chamilo IoT.owl](#) (attention: lorsque vous ferez l'enregistrement du fichier sur votre disque, ne modifiez pas l'extension qui doit rester `.owl`)
  3. Téléchargez et installez le logiciel Protégé :
    - a. Allez à l'URL <https://protege.stanford.edu/>
    - b. Cliquez sur le lien de téléchargement, suivant la plateforme utilisée et enregistrez le fichier sur votre disque
    - c. Extrayez l'archive,
    - d. Ouvrez le dossier obtenu
- NOTE : Sur la distribution Archlinux et ses dérivées, Protégé peut être installé à partir des dépôts : `# pacman -Sy protege`, il existe une note spécifique pour les systèmes linux. Merci de la télécharger depuis chamilo si vous en avez besoin. Pour le TP et le projet nous préconisons l'utilisation du système windows.
4. Lancez Protégé
    - a. Dans le dossier obtenu après l'extraction de l'archive Protégé (Protege-5.5.0 au 14/10/2021), lancez le script `run`
    - b. Sous Windows et Mac, faire double clic sur `run`
    - c. Sous Linux, ouvrez un terminal dans le dossier Protege-5.5.0 puis entrez successivement les commandes `chmod +x run.sh` et `sh run.sh`
  5. Ouvrez l'ontologie `IoT.owl`
    - a. Dans le menu **File** de Protégé choisissez l'option **Open...**,
    - b. Sélectionnez le fichier `IoT.owl` et ouvrez-le.

Pour les deux premières questions, vous vous concentrerez sur l'onglet **Entities** et ses "sous-onglets" (**Classes**, **Object properties**, **Data properties** ...). Notez aussi que les termes de l'ontologie sont des liens vers la définition correspondante. Ceci vous permet de naviguer d'un élément à l'autre, sans avoir à repasser à chaque fois par les panneaux de gauche. Notez enfin que le bouton < en haut à gauche vous permet de revenir en arrière dans votre navigation.



## Question 1

Sur la capture d'écran précédente, on voit par exemple que *Hygrometer* est une sous classe de la classe définie par les "choses" qui ont un composant (*hasComponent some Component*). Dit autrement, tout hygromètre a au moins un composant.

Pour chacun des éléments suivants, retranscrivez en français ce qu'en dit l'ontologie.

1. la classe *Device*
2. la propriété (*Object Property*) *hasComponent*
3. la propriété *isComponentOf*
4. la classe *Phone*
5. la classe *Smartphone*

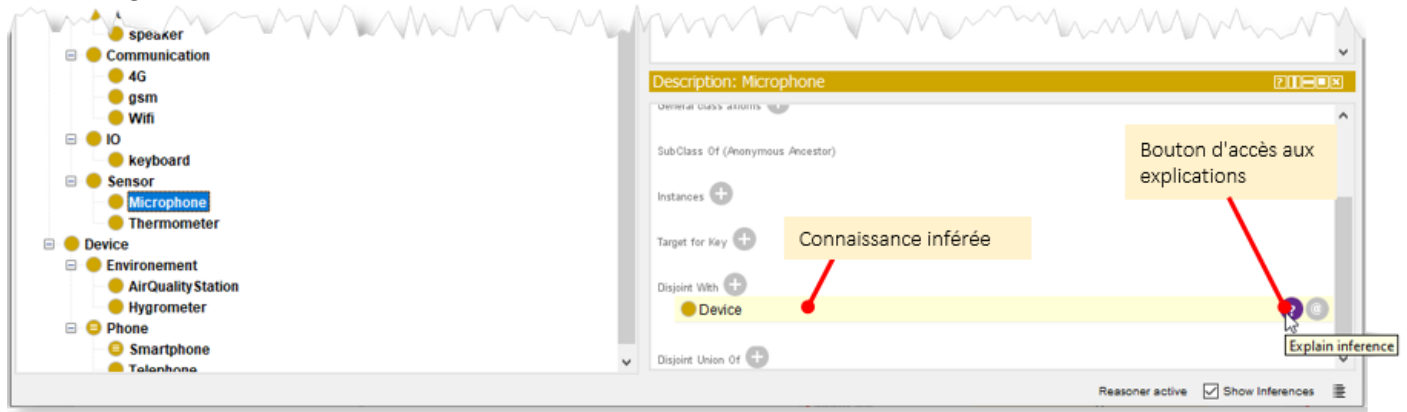
Important : contentez-vous de transcrire ce qui est *explicitement dit* par l'ontologie, pas ce que vous en déduisez ni ce que vous « devinez » en fonction des termes choisis.

## Question 2

Réglez les préférences du moteur d'inférence (raisonneur) : dans le menu *File > Préférences*, onglet *Reasoner*, cochez toutes les cases et validez ;

puis dans le menu *Reasoner*, cochez un raisonneur (*HerMiT* ou *FaCT* ou *Pellet*, selon ce qui est disponible). Activez ensuite le moteur d'inférence par le menu *Reasoner > Start reasoner* (ou *CTRL+R*).

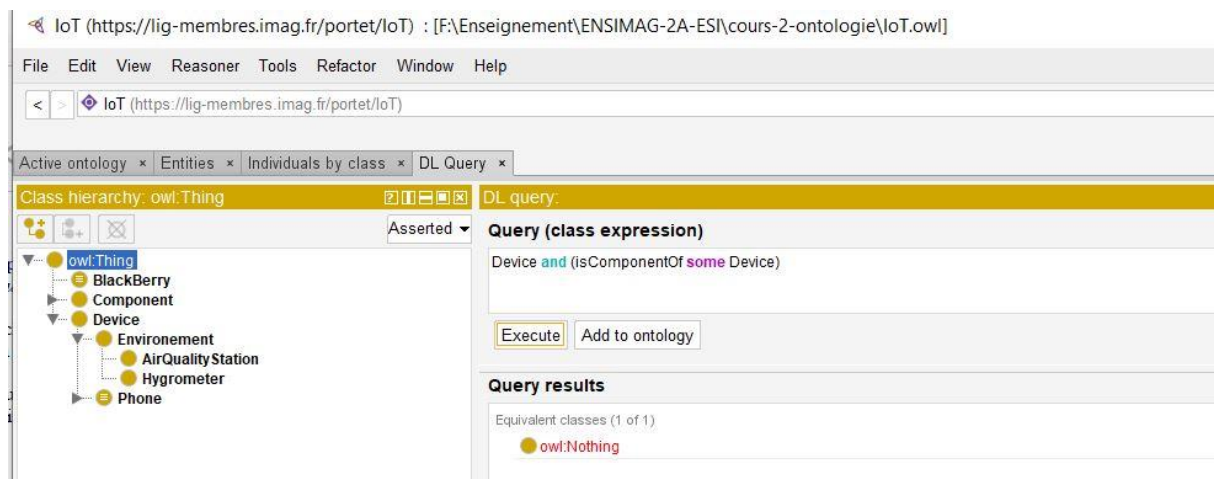
Explorez de nouveau les classes et notez que de nouvelles connaissances (sur fond jaune pâle) apparaissent. Pour chacune des classes et propriétés listées à la question précédente, décrivez en français les connaissances ajoutées par l'inférence sous les rubriques *Subclass of* et *Disjoint with*, et **expliquez** ces inférences (vous pouvez pour cela vous aider du bouton "?" situé à droite de chaque connaissance).



### Question 3

Ouvrez l'onglet **DL Query** (si il n'est pas visible en pouvez le faire apparaître en cochant **DL Query** dans le menu **Window > Tabs**. Dans le panneau de droite de cet onglet, assurez-vous que les cases **Direct superclasses**, **Equivalent classes** et **Direct subclasses** sont bien cochées.

Comme son nom l'indique DLQuery permet de faire des requêtes sur l'ontologie en utilisant les logiques de description. Par exemple, la capture d'écran suivante présente une requête qui définit (dans la syntaxe dite Manchester) la classe des Device qui sont aussi composants (isComponentOf) d'un Device. On voit que de telles entités ne peuvent exister car cette classe est équivalente à la classe vide owl:Nothing.



Le tableau ci-dessous résume les équivalences entre syntaxe OWL, DL (Description Logic) et Manchester. Au besoin, vous pouvez aussi consulter les exemples du document Manchester Syntax for OWL 1.1 section 6 ([https://protegewiki.stanford.edu/images/5/5f/Owled2008dc\\_paper\\_11.pdf](https://protegewiki.stanford.edu/images/5/5f/Owled2008dc_paper_11.pdf)) ou la rapide introduction suivante The Manchester OWL Syntax ([http://ceur-ws.org/Vol-216/submission\\_9.pdf](http://ceur-ws.org/Vol-216/submission_9.pdf))

| OWL Constructor | DL Syntax                  | Manchester         | OWL S. Example                     |
|-----------------|----------------------------|--------------------|------------------------------------|
| intersectionOf  | $C \sqcap D$               | <b>C AND D</b>     | Human <b>AND</b> Male              |
| unionOf         | $C \sqcup D$               | <b>C OR D</b>      | Man <b>OR</b> Woman                |
| complementOf    | $\neg C$                   | <b>NOT C</b>       | <b>NOT</b> Male                    |
| oneOf           | $\{a\} \sqcup \{b\} \dots$ | <b>{a b ...}</b>   | {England Italy Spain}              |
| someValuesFrom  | $\exists R C$              | <b>R SOME C</b>    | hasColleague <b>SOME</b> Professor |
| allValuesFrom   | $\forall R C$              | <b>R ONLY C</b>    | hasColleague <b>ONLY</b> Professor |
| minCardinality  | $\geq N R$                 | <b>R MIN 3</b>     | hasColleague <b>MIN 3</b>          |
| maxCardinality  | $\leq N R$                 | <b>R MAX 3</b>     | hasColleague <b>MAX 3</b>          |
| cardinality     | $= N R$                    | <b>R EXACTLY 3</b> | hasColleague <b>EXACTLY 3</b>      |
| hasValue        | $\exists R \{a\}$          | <b>R VALUE a</b>   | hasColleague <b>VALUE</b> Matthew  |

## TP-Ontologie

Dans l'onglet DL Query, pour chacune des questions suivantes écrivez (en syntaxe Manchester) et exécutez une expression qui permet de répondre à la question posée. Expliquez les réponses données (en vous aidant éventuellement du bouton "?").

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

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

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

## Question 4

Dans l'onglet *Individuals by class*, créez les instances suivantes :

Attention : On veut des instances (+ losange violet), PAS des sous-classes

- une instance de Device "fairphone\_1",
- une instance de Micro "micro\_1",
- une instance de GSM "gsm\_1".

Faites en sorte que "fairphone\_1" ait "micro\_1" comme composant et que "gsm\_1" soit un composant de "fairphone\_1". Pour cela on déclarera des **Object property assertions**. Rafraîchissez le raisonneur avec CTRL+R. Quelles connaissances le système a-t-il déduit sur ces instances ? Quel type de Device devient "fairphone\_1" ?

## Question 5

DBpedia est un dataset (graphe de connaissances) RDF (Resources Description Framework) construit à partir d'informations extraites automatiquement de Wikipedia. Ces ressources RDF sont bien plus facilement exploitables par des systèmes d'IA que les textes de Wikipedia (en effet, en RDF, les types et les propriétés des ressources sont définis de manière formelle à l'aide d'ontologies qui permettent de donner un sens (une sémantique) aux informations et d'effectuer des raisonnements sur celles-ci). Pour accéder aux données de DBpedia, nous utiliserons le langage de requêtes SPARQL (SPARQL By Example en propose une rapide introduction que vous pouvez consulter [https://www.iro.umontreal.ca/~lapalme/ift6281/sparql-1\\_1-cheat-sheet.pdf](https://www.iro.umontreal.ca/~lapalme/ift6281/sparql-1_1-cheat-sheet.pdf)).

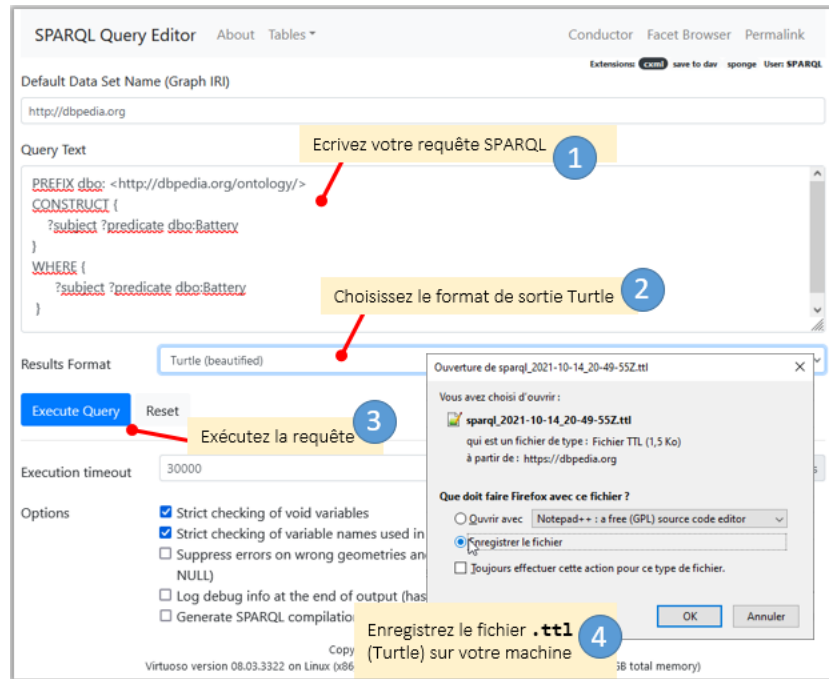
Dans ce qui suit, vous allez récupérer les différents types de batteries définis sur DBpedia

1. Dans votre navigateur allez sur <https://dbpedia.org/ontology/Battery> (ou bien sur <http://live.dbpedia.org/ontology/Battery>) et explorez les informations exposées.

Cette page présente les informations qui concernent la ressource <http://dbpedia.org/ontology/Battery>. Dans la suite on désignera cette ressource de manière abrégée par `dbo:Battery` en posant le namespace `dbo:` <http://dbpedia.org/ontology/>. On souhaite exploiter et récupérer une partie de ces informations pour enrichir notre ontologie avec les types de batteries.

2. Toujours dans votre navigateur allez sur <https://dbpedia.org/sparql> (ou bien sur <http://live.dbpedia.org/sparql/>), un *SPARQL endpoint* qui permet d'interroger DBpedia.
3. Exécutez la requête CONSTRUCT suivante en sélectionnant Results Format: Turtle pour télécharger de nouvelles informations sur `dbo:Battery` :

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



Si le fichier n'est pas téléchargé directement, faites Enregistrer sous "resultat-1.ttl" sur la page de résultats.

Une fois le fichiers résultats de votre requête enregistrée sur votre machine, vous allez revenir dans Protégé

4. Commencez par arrêter le raisonneur : menu **Reasoner > Stop reasoner**
5. Dans l'onglet **Active Ontology**, panneau **Imported Ontology**, cliquez sur **Direct Imports + -> Import an ontology contained in a local file -> Continue -> Browse** et sélectionnez le fichier .ttl téléchargé à l'étape précédente. Cliquez deux fois sur **Continue** pour accepter les choix par défaut et enfin **Finish**.
6. Dans l'onglet **Entities -> Classes** de Protégé, placez la nouvelle classe **dbo:Battery** sous la classe qui vous semble adéquate (glisser/déposer).
7. Dans l'onglet **Entities -> Individuals** déclarez que **fairphone\_1** a pour composant **dbpedia:Lithium-ion\_battery**
8. Dans le SPARQL endpoint de DBpedia, modifiez la requête **SELECT** de la question 3 pour récupérer les informations (triplets <?subject ?predicate ?object>) dont **dbo:Battery** est le subject.
9. Importez ces triplets dans Protégé en répétant l'étape 4 avec le nouveau fichier. ttl téléchargé
10. Dans l'onglet **Entities -> Classes** déclarez que la classe **Device** de notre ontologie est équivalente à **dbo:Device** qui vient de DBpedia. Réactivez le raisonneur (menu **Reasoner > Start reasoner**). Que constatez-vous ? Comment l'expliquer ? Comment le résoudre ?

### Pour aller plus loin : Enrichissement avec les données issues de Wikidata

Wikidata est un projet connexe à Wikipedia et basé sur le même fonctionnement de collaboration et de modération que Wikipedia. Au lieu de rédiger des articles en langue naturelles, les contributeurs créent des ressources RDF. Ces données servent entre autre à alimenter les nouvelles pages de Wikipedia (en particulier les info boxes) ce qui permet d'avoir une information cohérente entre les différentes versions d'une page dans des langues différentes (ce qui n'était pas garanti auparavant, puisque chaque page était un simple texte, modifier une donnée nécessitait de modifier toutes les pages)

Pour en savoir plus sur Wikidata, en particulier comment parcourir et interroger les données, vous pouvez vous rendre ici : <https://www.wikidata.org>

## Question 6

Ajoutez une classe “smartspeaker” et vérifiez la définition en vous aidant de requêtes DL et en créant des individus.

## Question 7

Proposez des extensions à l’ontologie pour pouvoir ajouter le type de device “AspirateurIntelligent”.