

## Quelques commentaires sur le projet

### Hérarchie de classes

- Ne pas confondre la relation est-un qui est la base de la hiérarchie de classes avec est-partie-de ou est-membre-de. Toujours se poser la question de savoir si une sous-classe en est bien une et si un individu est bien membre d'une classe.

Premier exemple :



Ici UFRMathInfo peut être sous-classe de Composante car l'UFR Math-Info **est** effectivement **une** composante. En revanche MasterGPI, MasterSIL, MasterSIRIS, MasterI3D et MasterSDSC ne peuvent être des sous-classes de UFRMathInfo car un master ne peut pas être une UFR. Une formation est rattachée à une UFR.

Il reste ensuite la question de savoir si UFRMathInfo a lieu d'exister comme classe : quels individus pourraient appartenir à cette classe ?

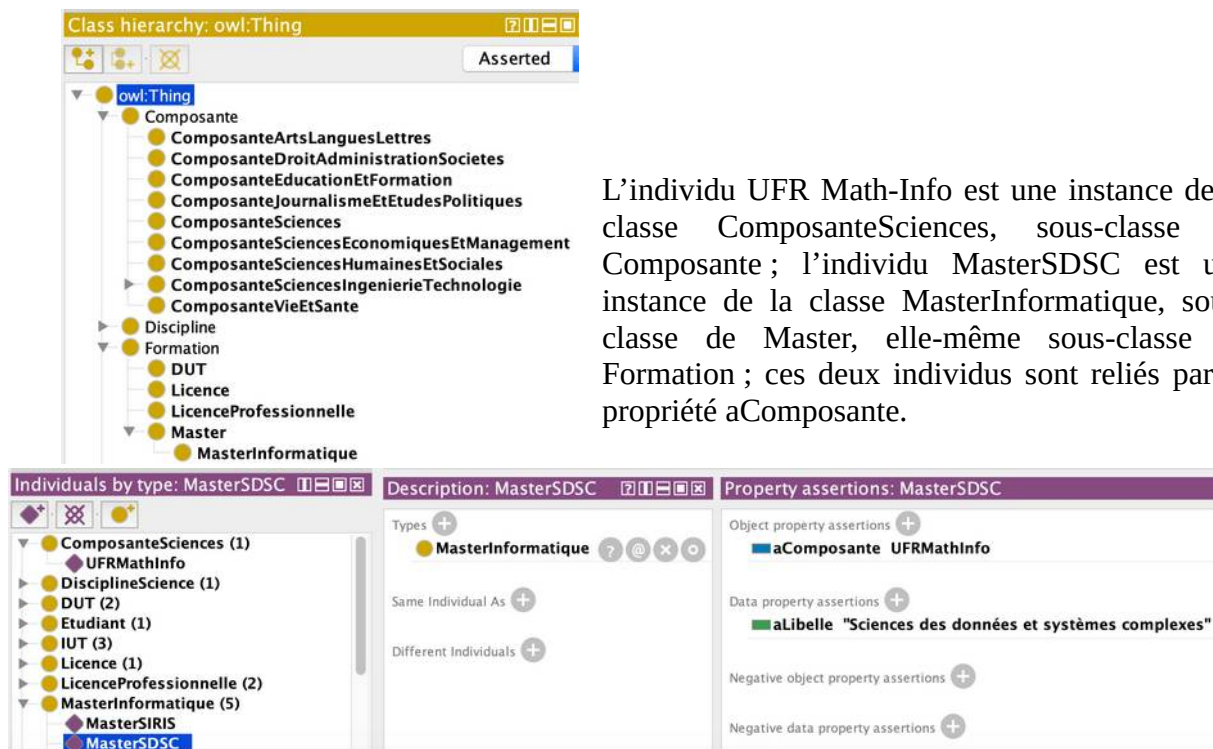
Deuxième exemple :



MasterGPI, MasterSIL, MasterSIRIS, MasterI3D et MasterSDSC sont cette fois des individus qui représentent ces 5 masters. Comme dans le premier exemple, un master n'est pas une UFR.

Ils ne peuvent donc pas non plus être des instances de la classe UFRMathInfo.

Une modélisation dépend de ce que l'on souhaite faire, mais dans tous les cas les exemples ci-dessus ne sont pas possibles. Une modélisation possible, suivant ce que l'on souhaite représenter, pourrait être la suivante



L'individu UFR Math-Info est une instance de la classe ComposanteSciences, sous-classe de Composante ; l'individu MasterSDSC est une instance de la classe MasterInformatique, sous-classe de Master, elle-même sous-classe de Formation ; ces deux individus sont reliés par la propriété aComposante.

- Penser à spécifier que des classes sont disjointes lorsque c'est le cas
- Vérifier que votre hiérarchie est cohérente et que le raisonneur ne modifie pas la hiérarchie d'une manière incohérente

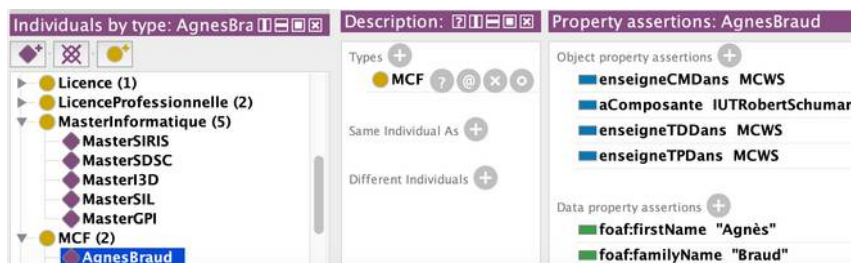
## Relations ternaires

- Il peut être pertinent d'utiliser des relations ternaires pour une modélisation plus élégante et efficace.

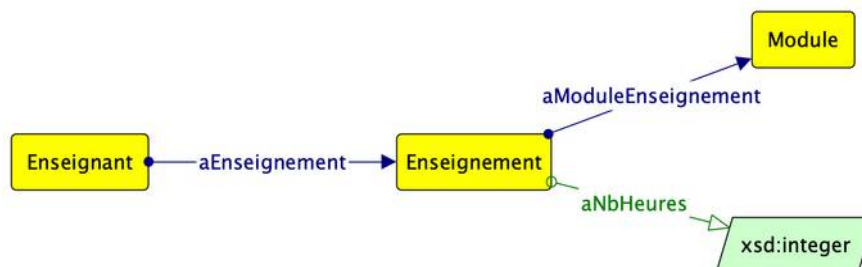
Considérons l'affectation des heures CM/CI/TD/TP aux enseignants. On peut utiliser la modélisation suivante, avec les propriétés `enseigneCMDans`, `enseigneCIDans`, `enseigneTDDans` et `enseigneTPDans` sous-propriétés de `enseigneDans`



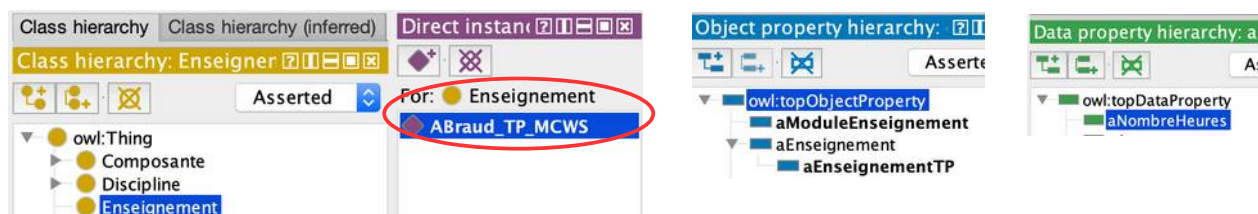
Ce qui donne sous Protégé pour l'exemple Agnès Braud dispense des CM, TD et TP dans le module MCWS



Je pourrais avoir envie de préciser aussi le nombre d'heures, je peux le faire ainsi (avec `aEnseignementTP` sous propriété de `aEnseignement`)



Cela se traduit par l'ajout de la classe `Enseignement` et des propriétés `aEnseignementTP`, `aModuleEnseignement` et `aNbHeures`



On peut alors indiquer que Agnès Braud enseigne 10h de TP dans le module MCWS (avec les éléments en rouge)





On pourrait également rajouter le type d'enseignement (CM/CI/TD/TP) de cette manière plutôt que de l'intégrer dans le nom de la propriété.

## Utilisation de vocabulaires existants

- Bien vérifier comment les éléments repris d'un vocabulaire existant sont décrits dans la spécification de ce vocabulaire. Eventuellement, s'il y a équivalence avec un élément d'un autre vocabulaire, vérifier la spécification dans ce vocabulaire afin d'affiner la compréhension. On ne peut pas le détourner de son sens, c'est alors un élément différent qui n'a pas à être lié au vocabulaire.

Ainsi title de foaf correspond à la civilité, et ne peut pas être le titre d'un livre par exemple.



## Individus

- Ne pas utiliser le nom d'un individu comme ayant un sens, c'est un identifiant. En revanche, c'est bien lui qui permet de distinguer deux individus qui pourraient avoir la même description, à supposer qu'ils sont bien indiqués comme différents (car dans le monde ouvert, rien ne dit que deux individus sont différents, sauf si cela est indiqué explicitement)
- Un individu devrait pouvoir se créer à la racine de la hiérarchie de classe et se classer dans les sous-classes lui correspondant grâce à leur description.

## SPARQL

- Utiliser les commandes adaptées : ne pas utiliser un GROUP BY pour un DISTINCT.
- S'assurer des types de données quand c'est nécessaire.