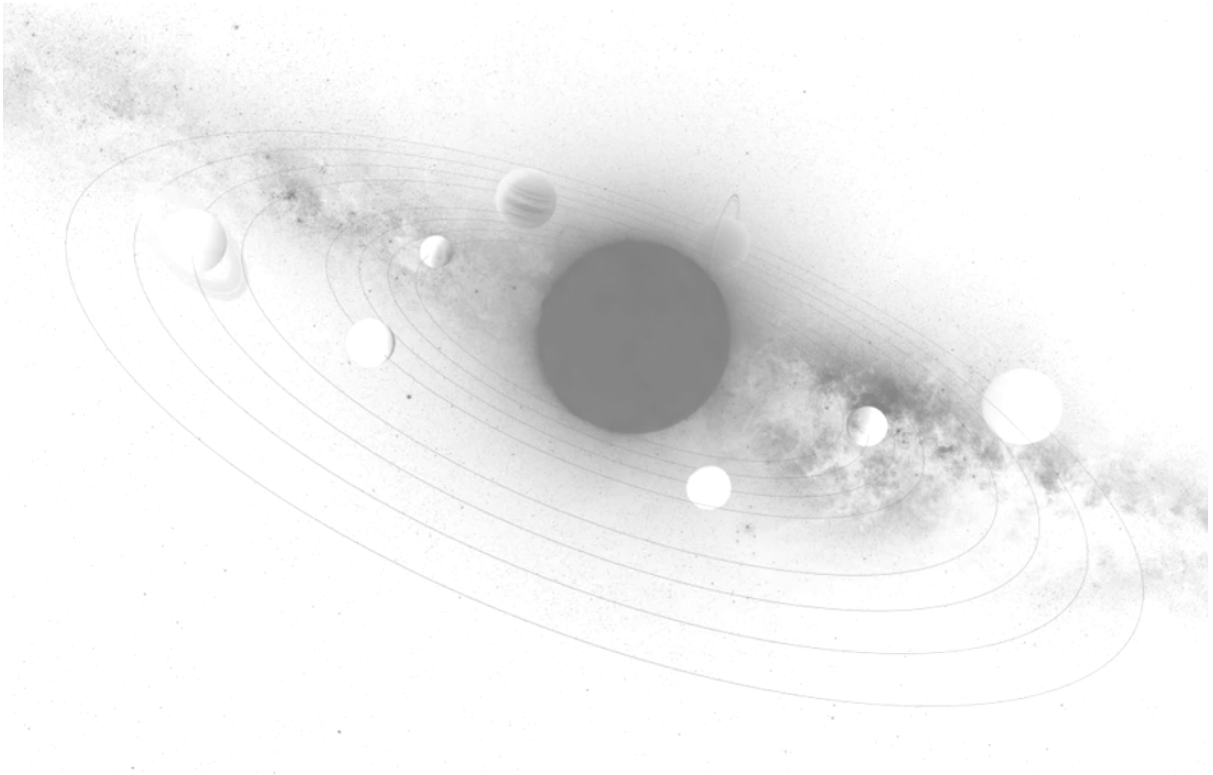
Le système solaire

Ontologie Protégé - IA301 - MS IA



Sami Allani
Marie-Elisabeth Campo
Samuel Flandrin
Abdelkarim Moussaid
Léa Papillon
Louis-Gabriel Pouillot

Sommaire

Introduction

L'ontologie autour du Système Solaire présentée dans ce document a été créée dans le cadre de Travaux Pratiques du module IA301 "Logique et IA Symbolique", du MS IA.

An ontology is "a formal specification of a shared conceptualization (Gruber 1993), a formalism to define concepts, individuals, relationships and constraints (functions, attributes) within a domain."

Une ontologie est une représentation formelle d'un champ de connaissances donné. Elle est structurée en "concepts" hiérarchisés, éventuellement liés par des relations, et dont il est possible d'instancier des "individus".

- Concept : définition formelle d'une agrégation de choses
- Individus : instance d'un concept
- Relations : lien entre concepts / individus

La représentation d'un champ de connaissance sous forme d'une ontologie présente plusieurs intérêts.

D'abord, elle permet aux experts du domaine de s'accorder sur des définitions communes (caractéristiques intrinsèques des concepts) et sur un vocabulaire uniforme (standardisé), qui favorisent les collaborations. Une fois définie, cette ontologie constitue une représentation pérenne du domaine, qui demeure lorsque l'expert part.
Ces représentations formelles peuvent ensuite servir de support à des raisonnements logiques (déduction, abduction…).

Dans le cadre de ce projet, nous avons choisi de représenter formellement les entités du Système Solaire (planètes, étoiles, satellites, astéroïdes, objets trans-neptuniens, comètes, etc.). Nous présentons ici le modèle que nous avons créé sur Protégé (concepts, relations, instances), et les fonctionnalités d'inférence qui nous ont permis de compléter automatiquement l'ontologie. Nous verrons également quelques exemples de requêtes qui nous ont permis d'exploiter le modèle.

I- Le modèle de données

Les classes

- owl:Thing
    - Astres
    - Astéroïdes
    - Ceinture_de_Kuiper
        - Objet-Transneptuniens
    - Etoile
    - Nuage_de_Oort
        - Comètes
    - Planètes
    - Satellites
    - ObjetsArtificiels
    - Sonde

Les propriétés des objets

- owl:topObjectProperty
    - aCommeSatellite
    - aMoinsDeSatellitesQue
    - aPlusDeSatellitesQue
    - aVisité
    - aEtéVisitéPar
    - estDeLaMêmeCouleurQue
    - estEnRotationSynchroneAvec
    - estEnRésonanceOrbitaleAvec
    - tourneAutourDe
    - ALancé
    - AétéLancépar
    - aRéaliséUneMissionVers
    - aEteObjectifDe

Les propriétés des données

- owl:topDataProperty
    - aUneActivitéInterne
    - couleur
    - densité
    - distance
    - développeUneQueue
    - engendreDesRéactionsNucléaires
    - forme

        - irrégulière
        - sphérique
    - masse
    - nature
        - gazeuse
        - tellurique
    - possèdeDesAnneaux
    - possèdeUneAtmosphère
    - périodeDeRotation
    - périodeDeRévolution
    - rayon

Les individus

Albion	Giotto	Mercure	Soleil
Callisto	Hale-Bopp	Mimas	Stardust
Cassini_Huygens	Hayabusa	Miranda	Tchoury
Change'e	Haykutake	Near_Schoemaker	Tempel1
Charon	Hygie	Neowise	Terre
ComèteDeHalley	Hyperion	Neptune	Titan
Cérès	Io	New_Horizon	Titania
Dawn	Itokawa	Obéron	Triton
DeepImpact	Juno	Pallas	Uranus
Deimos	Jupiter	Phobos	Vega
Encelade	Luna	Pioneer	Venera
Epiméthée	Lune	Pluton	Venus
Eris	Magellan	Protée	Vesta
Eros	Makémaké	Rhéa	Viking
Europe	Mariner	Rosetta	Voyager1
Galileo	Mars	Saturne	Voyager2
Ganymède	Maven	Sedna	Wild

Les relations

Les propriétés listées précédemment ont les caractéristiques suivantes :

aVisité : est la propriété inverse de **aEtéVisitéPar**. Aucune des deux n'est réflexive, ni symétrique. La transitivité n'a pas de sens.

aPourSatellite : inverse de t**ourne AutourDe**. De la même façon, ces relations ne sont ni symétriques, ni réflexives, ni transitives.

tourneAutourDe : pas transitif

estEnRésonanceOrbitaleAvec : signifie que les orbites sont parcourues dans des temps qui sont en rapport entiers entre elles. Cette relation est par nature symétrique mais pas transitive ni euclidienne : par exemple Io est en résonance avec Ganymède et Europe, mais Europe et Ganymède ne sont pas en résonance ensemble, ou du moins pas de façon significative.

estEnRotationSynchroneAvec : signifie que la période de révolution autour de la planète est égale à la période de rotation du satellite sur lui-même, ce qui implique un verrouillage gravitationnel où le satellite présente toujours la même face à la planète, du aux effets de marée : relation pas symétrique, le satellite présente la même face à la planète mais la planète ne présente pas la même face au satellite (exemple Terre/lune). Pas réflexive, ni transitive.

estDeLaMêmeCouleurQue : symétrique, transitif. Pas reflexif, sinon s'applique à toutes les classes du modèle, y compris celle pour qui la notion de couleur n'a pas de sens (les pays par exemple)

aPlusDeSatellitesQue : inverse de **aMoinsDeSatellitesQue,** transitif mais pas symétrique. C'est une relation d'ordre.

ALancé : inverse de **AétéLancépar**

Nous aurions aimé pouvoir composer les relations, comme ALancé et Avisité, pour pouvoir relier une astre à un pays porteur d'une mission (par exemple USA a lancé Juno, Juno a visité Jupiter, donc USA a visité Jupiter), mais nous n'avons pas trouvé comment faire.

II- Les inférences

Le moteur d’inférence

Un moteur d'inférence permet de conduire des raisonnements logiques et de dériver des conclusions à partir d’une base de connaissances ou de faits.

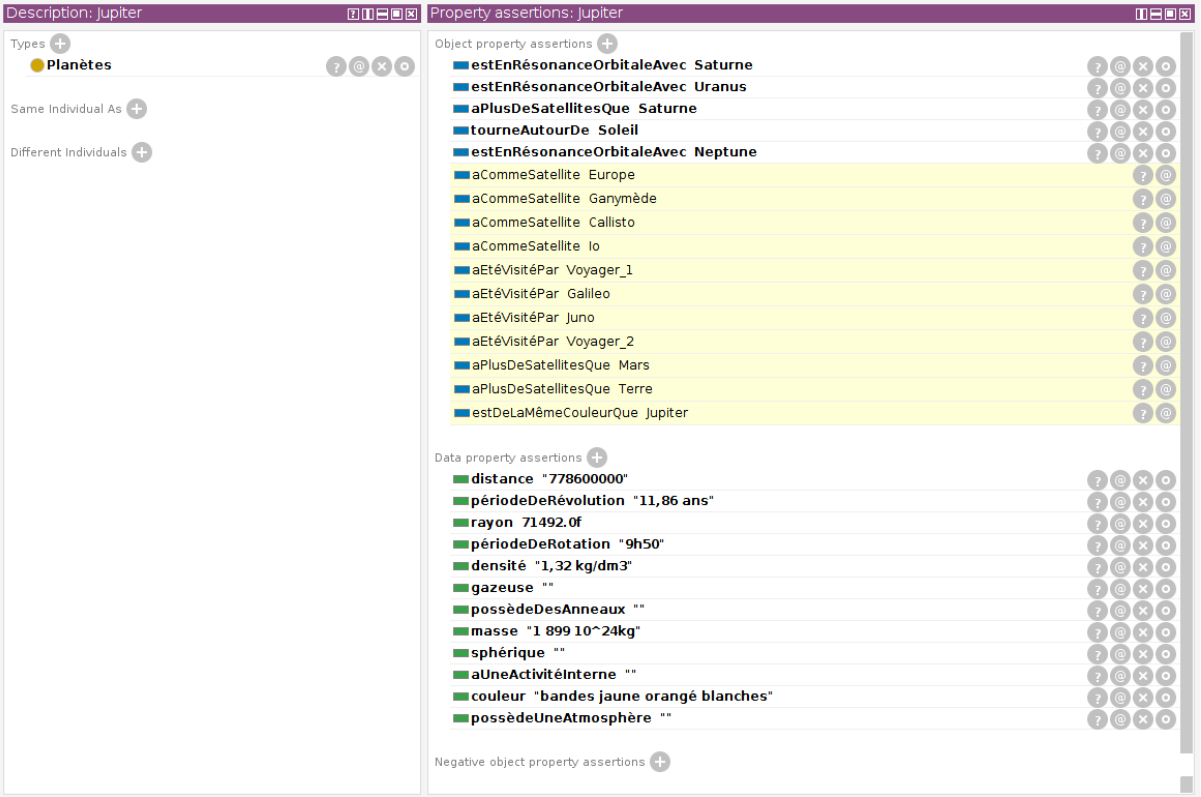
Protégé embarque plusieurs moteurs d’inférence par défaut. Nous avons utilisé le moteur Hermit 1.4.3.456 dans le cadre de ce TP.

En s’appuyant sur les connaissances ajoutées manuellement et sur les propriétés des relations présentées précédemment, l’algorithme de simulation des raisonnements déductifs a pu réaliser plusieurs inférences qui sont présentées ci-dessous :

Exemples d’inférences

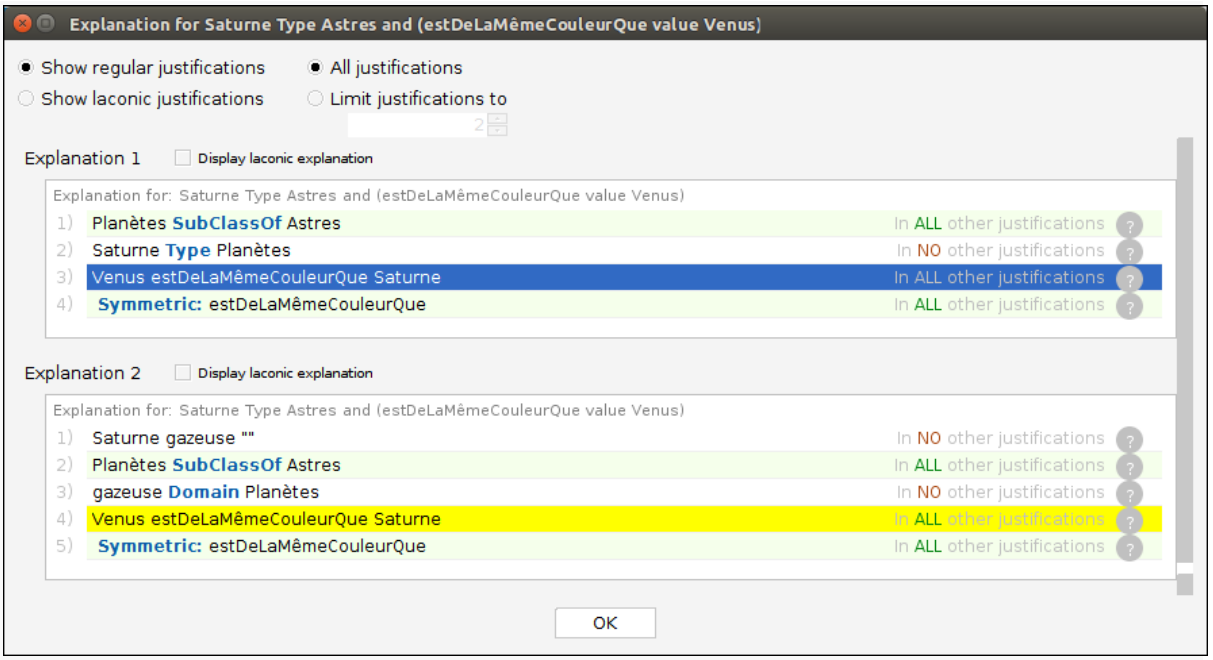
Inférences pour Jupiter

Nous illustrons les inférences réalisées pour Jupiter dans la vue ci-dessous.
Nous voyons que Jupiter a plus de satellites que Mars car Jupiter en a plus que Saturne, et Saturne en a plus que Mars. Les satellites de Jupiter sont également déduits car nous avons renseigné pour chaque satellite autour de quelle planète il tourne, ce qui permet d’inférer lesquels Jupiter a comme satellites. De même, nous avons renseigné pour chaque sonde quel corps elle a visité, ce qui nous permet d’inférer par quelles sondes cette géante gazeuze a été visitée..

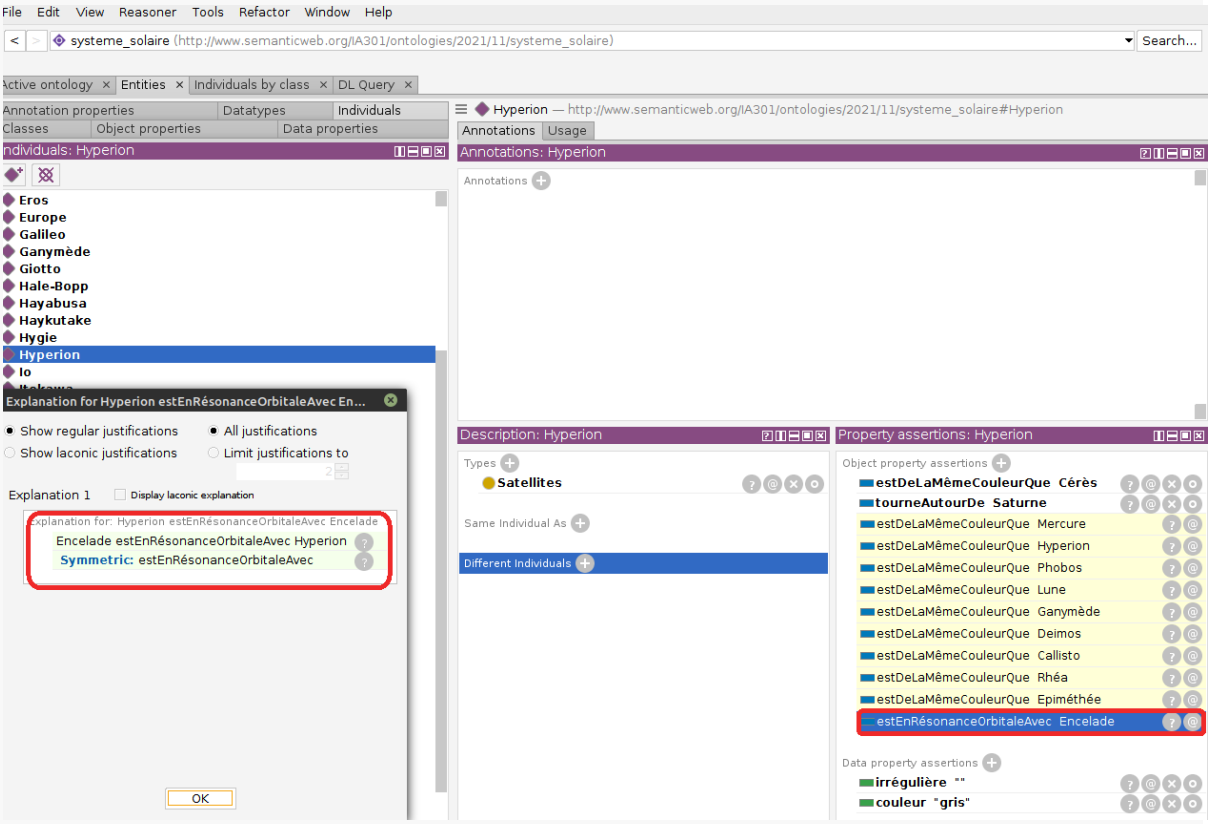


Vue Explanation d'une inférence

Nous mettons en évidence les raisons du résultat de ces inférences sur la fenêtre *Explanation* pour une requête Descriptive Language (DL Query) telle que *"Astres and estDeLaMêmeCouleurQue value Venus*". Saturne est de la même couleur que Vénus car on a dit que Titan était de la même couleur que Vénus, et que Saturne.



Par symétrie également on obtient que Hypérion est en résonance orbitale avec Encelade puisqu'on a indiqué qu'Encelade était en résonance avec Hypérion. Ce qu'on peut visualiser ci-dessous.

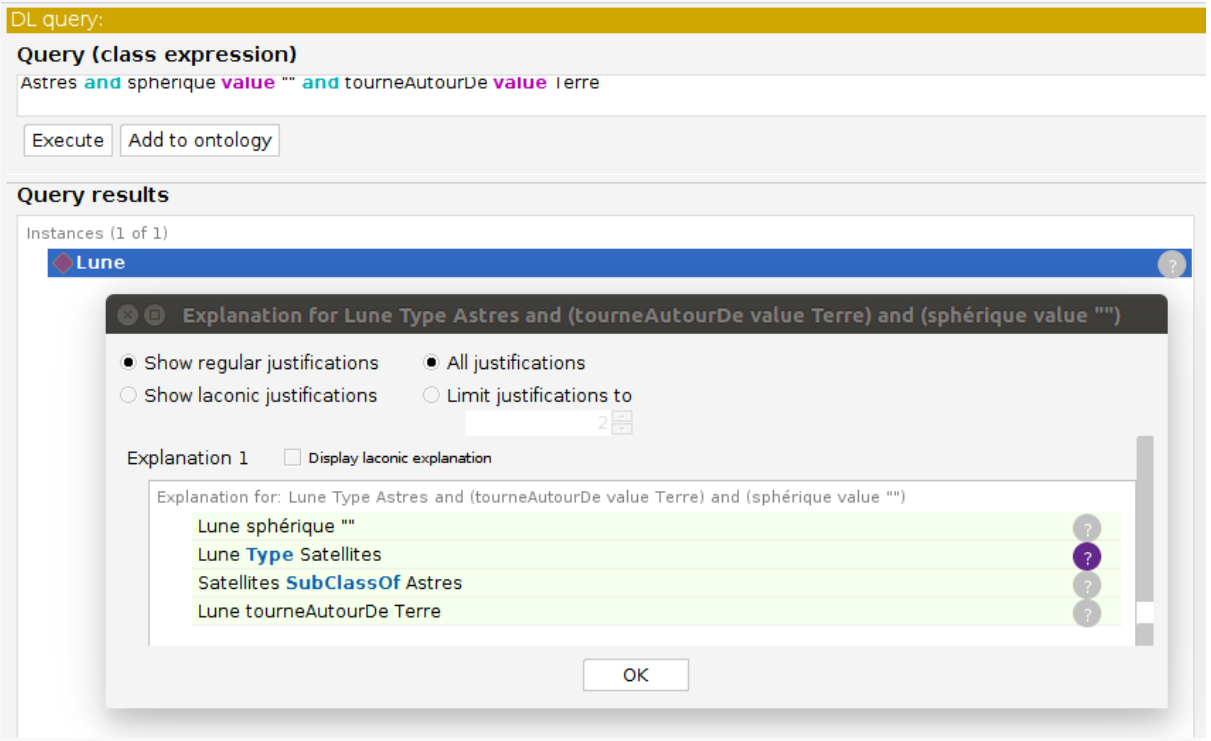


III- Requêtes

Avec le moteur d'inférence nous avons utilisé l'outil de requêtes DL Query pour réaliser plusieurs requêtes présentées dans cette partie.

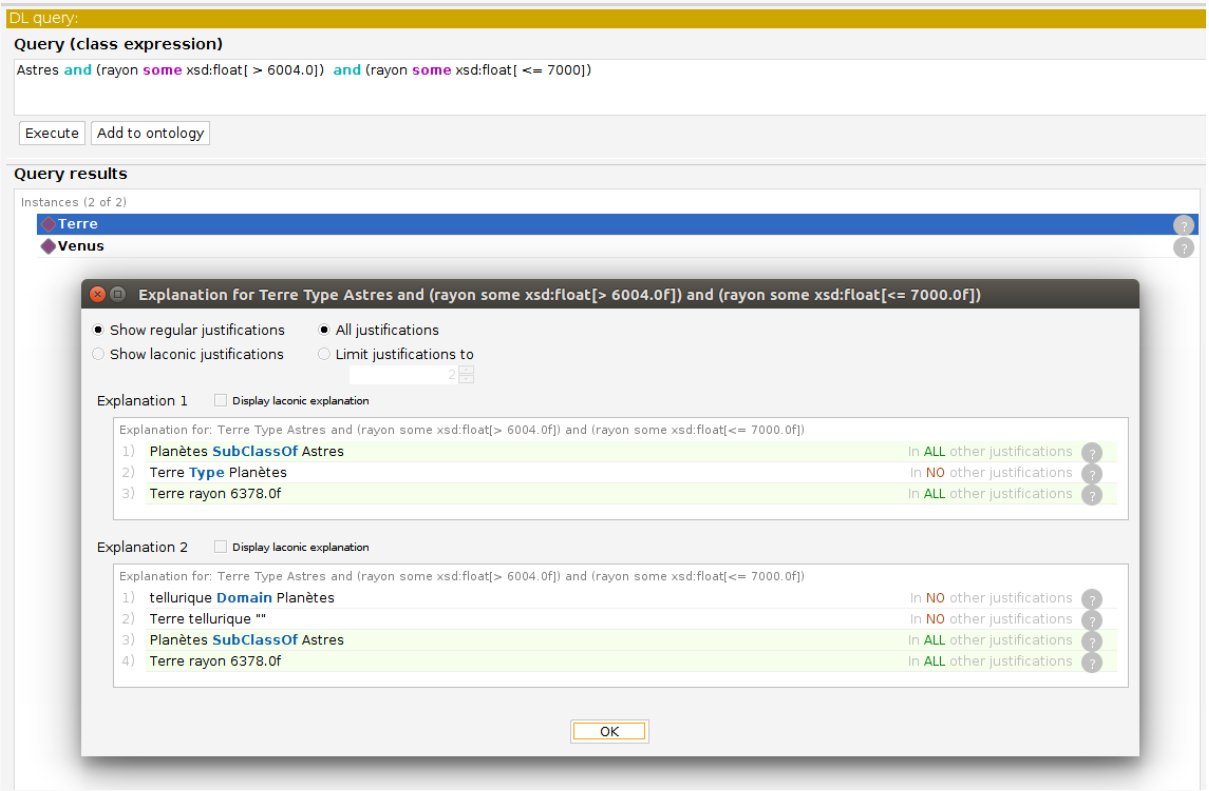
Retrouver la Terre et Vénus

Requête : Astres and sphérique value "" and tourneAutourDe value Terre



Utiliser des valeurs numériques

Requête avec des valeurs numériques sur le rayon : Astres and (rayon some xsd:float[> 6004.0]) and (rayon some xsd:float[<= 7000])

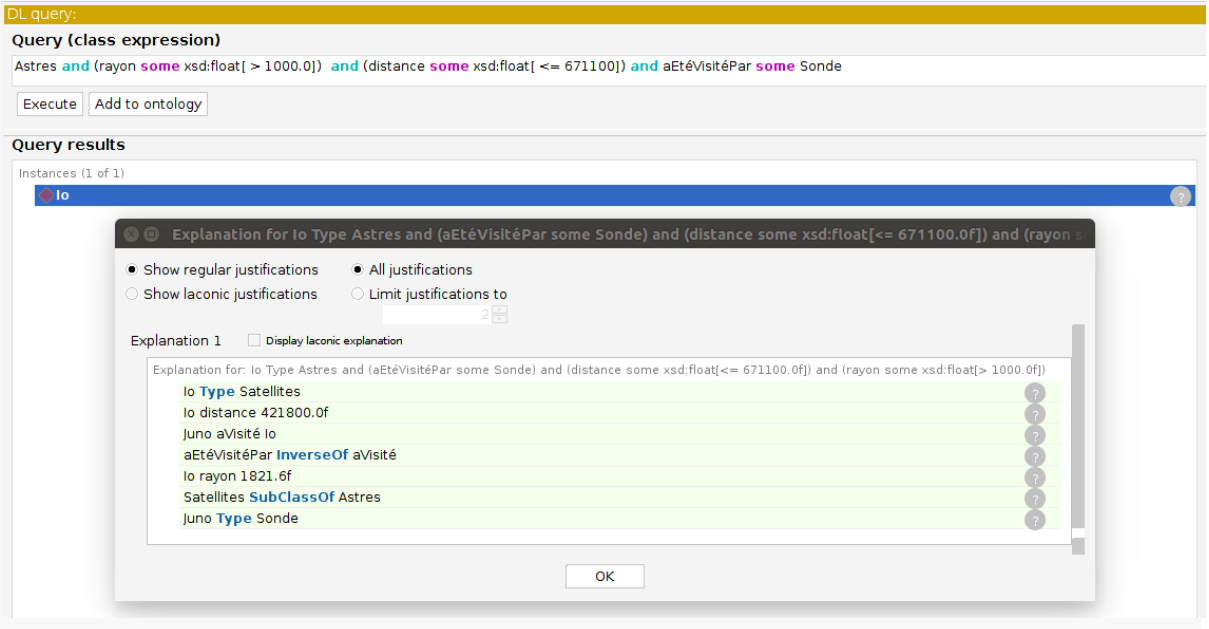


Une requête plus complexe

Requête : Astres and (rayon some xsd:float[> 1000.0]) and (distance some xsd:float[<= 671100]) and aEtéVisitéPar some Sonde

Nous avons retrouvé Io par son rayon, sa distance à Jupiter autour de laquelle elle

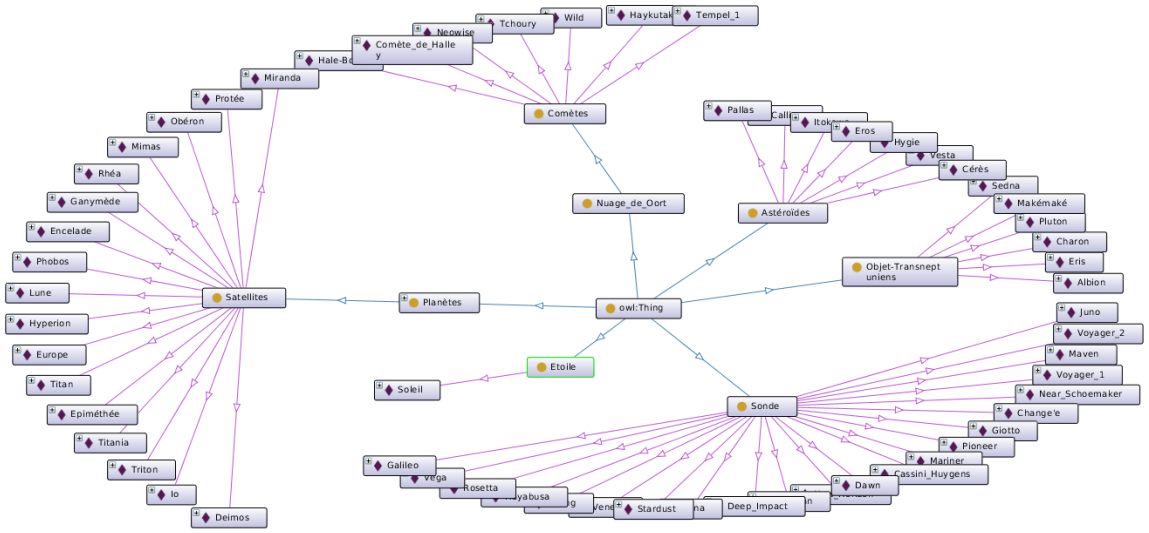
gravite et du fait qu'elle a été visitée par une sonde, Juno.



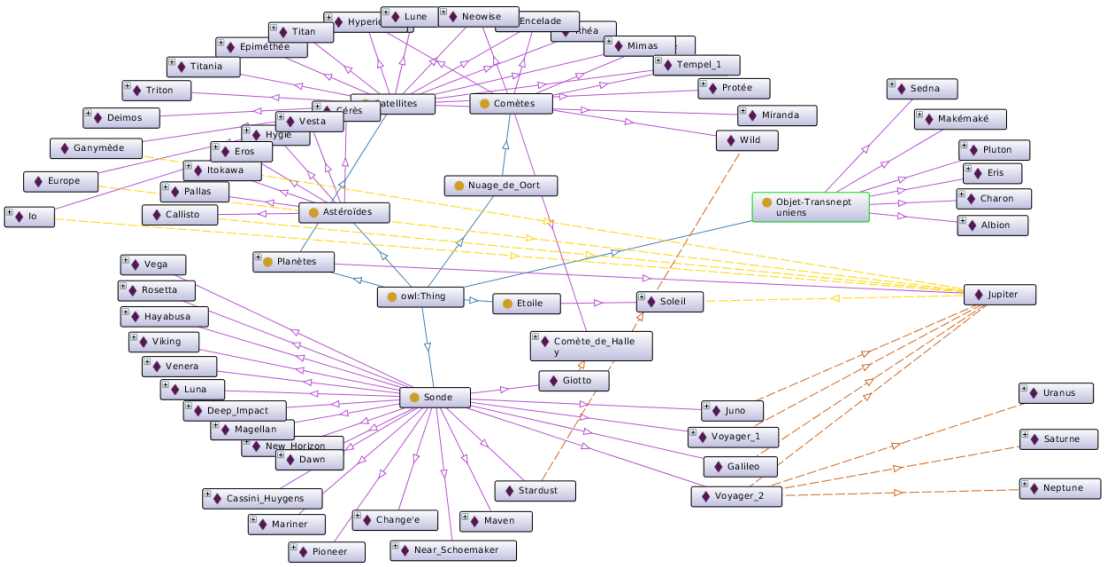
IV- Visualisations

Pour réaliser les visualisations de cette partie nous avons utilisé Ontograf qui est intégré à Protégé.

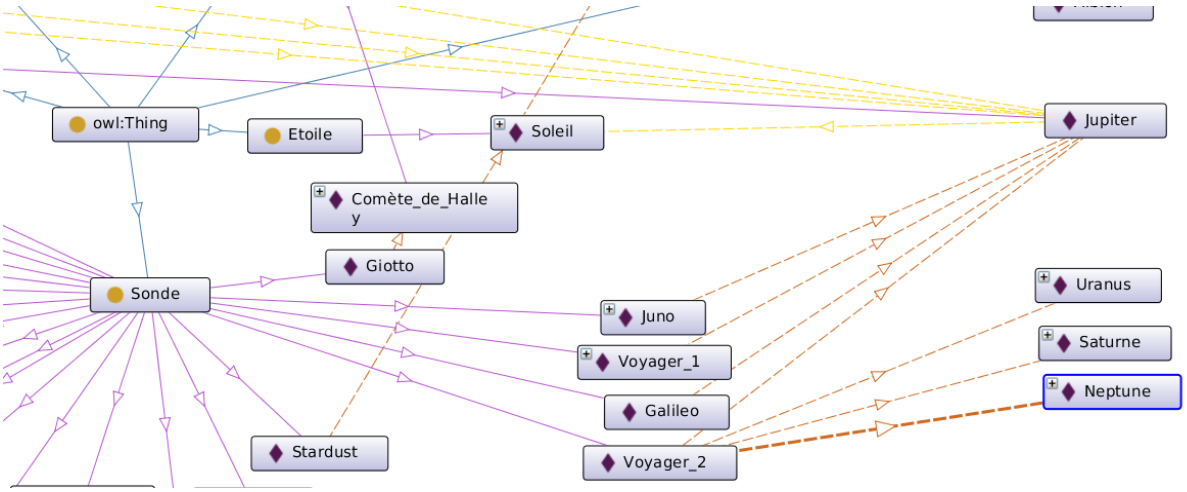
Visualisation des classes (vue d'ensemble)



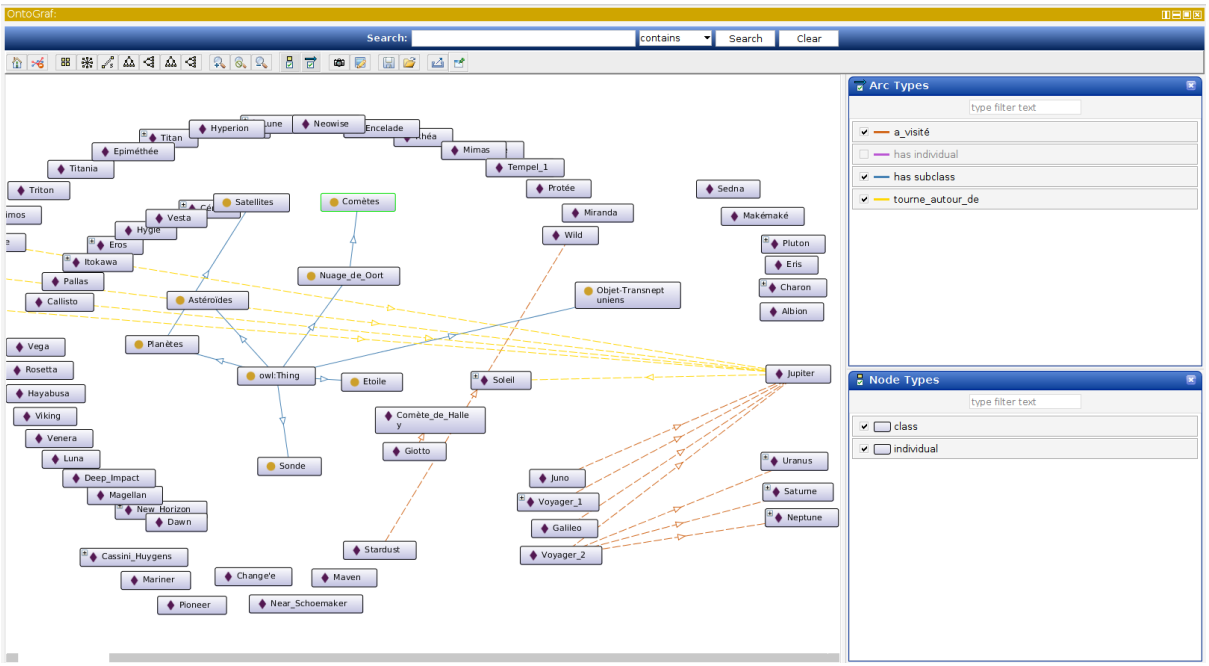
Visualisation des classes (avec quelques object properties)



Visualisation des classes et object properties (zoom)



Visualisation d'une sélection de arc types et node types



Conclusion

En conclusion, ce projet nous a permis de nous initier au logiciel de construction d'ontologie "Protégé". Nous avons trouvé le logiciel assez intuitif, complet et puissant (même si nous avons remarqué quelques bugs lors de l'utilisation). Pour la visualisation nous avons expérimenté OntoGraf.

Le sujet nous a confronté aux difficultés de la construction d'une ontologie (identification et définition des concepts et de leurs relations, insertion fastidieuse des données…) et nous a permis de voir l'intérêt des fonctionnalités fournies par Protégé : inférence de connaissances exploitation des données par requêtage, etc. Nous avons également pu éditer le fichier .owl à la main pour accélérer les mises à jours des données.

Ce travail a également été l'occasion pour nous d'en apprendre plus sur le système solaire.

Livrables

- le présent rapport
- l'ontologie owl *systeme_solaire.owl*
- le jupyter notebook *DecisionTree.ipynb* (1ère partie du TP)

Versions de logiciels et bibliothèques

- https://protege.stanford.edu/ v. 5.5.0
- Hermit Reasoner v. 1.4.3
- https://www.graphviz.org/ v. 2.49.3