

# Ontologie D&D

## Projet final de Web Semantique

Cédric Pendville, Quentin Juilland

### 1 Introduction

Le projet consiste à modéliser une ontologie dans l'univers de Donjons & Dragons (D&D). Cette ontologie a été développée pour représenter les éléments clés de cet univers, tels que les races, classes, factions, personnages, et leurs relations. Elle utilise le format RDF pour garantir une compatibilité avec les technologies du Web sémantique et intègre un ensemble de requêtes SPARQL pour extraire des informations pertinentes.

### 2 Conception de l'ontologie

L'ontologie modélise les concepts suivants :

- *Classes principales*

- Alignement
- Attribut
- Capacité
- Classe
- Compétence
- Créature
- EmplacementSort
- Faction
- Lieu
- Objet
- Personnage
- Quête
- Race
- Région
- Sort

- *Relations principales*

- `rdfs:subClassOf`
- `aComposant`
- `aDuree`
- `aEcoleMagie`
- `aEmplacementSort`
- `aModeLancement`
- `aNiveau`
- `aPortee`

- aPourAlignement
- aPourAllies
- aPourCapacite
- aPourClasse
- aPourEnnemis
- aPourLeader
- aPourProprietaire
- aPourRace
- aPourTerritoire
- aTempsIncantation
- connaitSort
- donneBonus
- hasSpeed
- maitriseCompetence
- seTrouveA

### 3 Données peuplées

L'ontologie inclut des instances fictives pour simuler un univers D&D :

- **Races** : Tieffelin, Elfe, Humain, etc., avec leurs vitesses et bonus.
- **Classes** : Assassin, Guerrier, Barde, Magicien, etc.
- **Factions** : LaGardeRoyale, L'Ordre des Ombres, etc.
- **Personnages** : Exemple de personnages alignés sur des factions spécifiques.

### 4 Requêtes SPARQL

Les requêtes SPARQL suivantes ont été implémentées pour interroger l'ontologie :

1. Récupérer tous les personnages et leur alignement

```
SELECT ?personnage ?alignement
WHERE {
    ?personnage a :Personnage .
    ?personnage :aPourAlignement ?alignement .
}
```

Montre les requêtes de base.

2. Trouver les classes disponibles pour les personnages

```
SELECT DISTINCT ?classe
WHERE {
    ?classe rdfs:subClassOf+ :Classe .
    FILTER NOT EXISTS {
        ?subClass rdfs:subClassOf ?classe .
    }
}
```

Montre l'utilisation de filtres

## 3. Lister les factions et toutes leurs informations

```

SELECT DISTINCT ?faction ?type ?alignement ?allies
?ennemis ?leader
WHERE {
?type rdfs:subClassOf :Faction .
?faction a ?type .
?faction :aPourAlignement ?alignement .
OPTIONAL { ?faction a ?type . }
OPTIONAL { ?faction :aPourAllies ?allies . }
OPTIONAL { ?faction :aPourEnnemis ?ennemis . }
OPTIONAL { ?faction :aPourLeader ?leader . }
OPTIONAL { ?faction :aPourTerritoire ?territoire . }
}

```

Montre l'utilisation d'OPTIONAL

## 4. Identifier les races et leur vitesse

```

SELECT ?race ?speed
WHERE {
    ?race rdfs:subClassOf :Race .
    ?race :hasSpeed ?speed .
}
ORDER BY DESC(?speed)

```

Montre l'utilisation de ORDER

## 5 Règle SWRL

Une règle SWRL a été définie pour automatiser une logique simple, si une race donne un bonus au Charisme et a une vitesse supérieure à 30, elle est considérée comme "Charismatique et rapide".

```

dnd:Race(?r) ^ dnd:donneBonus(?r, dnd:Charisme) ^ dnd:hasSpeed(?r, ?s)
^ swrlb:equal(?s, 30) -> dnd:CharismatiqueEtRapide(?r)

```

Malheureusement, une fois dans Protégé, la règle ne fonctionne pas, pour une raison qui nous est encore inconnue.

## 6 Jupyter Notebook

Le notebook inclut le chargement de l'ontologie, l'exécution des requêtes SPARQL avec la librairie `rdflib`. Les résultats sont ensuite traités pour apparaître de manière lisible.

## 7 Conclusion

Ce projet démontre comment une ontologie peut structurer et interroger un univers complexe comme celui de D&D. Il illustre l'intégration des technologies du Web sémantique pour représenter et exploiter des données hiérarchiques et interconnectées.

Les choix de modélisation ont permis une représentation claire des relations hiérarchiques et des propriétés spécifiques des entités. Cependant, des améliorations pourraient inclure l'ajout d'annotations sémantiques plus riches ou des règles supplémentaires pour des analyses automatisées.