

REPUBLIQUE DU SENEGLAL



Un Peuple – Un But – Une Foi

MINISTÈRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE

UNIVERSITE GASTON BERGER DE SAINT-LOUIS



U.F.R DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

MASTER 2 INFORMATIQUE - GDIL

Rapport de projet Web Sémantique

Titre : Description sémantique des ressources de l'enseignement supérieur et de la recherche

Présenté par :

Mouhamadou GUEYE
Daouda FICKOU
Lala DIALLO
Sokhna Binetou DIENE
Hameth DIALLO

Année académique: 2024/2025

Table des matières

1. Contexte	2
2. Problématique	2
3. Travail Réalisé.....	3
3.1 Choix de Modélisation (RDFS)	3
3.2 Contraintes OWL Utilisées.....	4
3.3 Exemples d'Inférences	4
3.4 Classification Automatique (Enseignant-Chercheur).....	11
4. Perspectives	11

1. Contexte

Les établissements d'enseignement supérieur et de recherche, tels que l'Université Gaston Berger (UGB) ou l'Université Assane Seck de Ziguinchor (UASZ), produisent et exploitent quotidiennement un volume considérable de ressources hétérogènes. Cet écosystème complexe regroupe à la fois des :

- **Ressources académiques** (formations, unités d'enseignement, maquettes pédagogiques),
- **Ressources humaines** (enseignants, chercheurs, doctorants, étudiants, personnel administratif),
- **Ressources scientifiques** (laboratoires, projets de recherche, publications),
- **Infrastructures et équipements** (bâtiments, salles de cours, plateformes numériques, matériel scientifique).

Actuellement, la gestion de ces informations est fragmentée. Elles sont dispersées dans des systèmes d'information cloisonnés (bases de données de scolarité, portails de recherche, fichiers bureautiques, sites web institutionnels) qui utilisent leurs propres formats et conventions sans véritable standard commun. Cette absence de lien structurel empêche une vision globale et freine l'échange de données.

C'est ici qu'intervient le Web sémantique. Portée par les standards du W3C (RDF, RDFS, OWL, SPARQL), cette approche offre un cadre technologique pour décrire formellement les ressources et leurs relations. Elle permet de passer d'un simple stockage de données à une véritable représentation de la connaissance, rendant les informations compréhensibles par les machines. Ce projet s'inscrit précisément dans cette démarche : il vise à concevoir une ontologie RDFS/OWL capable d'unifier la description des ressources universitaires pour en faciliter le partage et l'exploitation intelligente.

2. Problématique

Malgré la richesse des données disponibles, leur exploitation se heurte à plusieurs verrous majeurs liés à l'architecture actuelle des systèmes d'information universitaires :

1. **Hétérogénéité et cloisonnement** : Le système gérant les inscriptions administratives ne communique pas avec celui des laboratoires de recherche. Ainsi, il est difficile de relier automatiquement un doctorant à son sujet de thèse ou de savoir si un enseignant est actif dans un projet de recherche spécifique.
2. **Absence d'interopérabilité sémantique** : Les termes utilisés varient d'un système à l'autre sans mécanisme formel pour les aligner. Rien ne garantit qu'une "UE" dans le logiciel de l'emploi du temps correspond à la même entité qu'un "Cours" dans la plateforme pédagogique.
3. **Impossibilité de raisonnement automatique** : Les bases de données classiques ne permettent pas de déduire de nouvelles connaissances. Par exemple, elles ne peuvent

pas inférer automatiquement qu'un individu qui enseigne et qui est membre d'un laboratoire est de fait un "Enseignant-Chercheur", ni détecter l'incohérence si une personne est déclarée à la fois étudiant en Licence et Professeur titulaire.

4. **Manque de standardisation** : Sans l'usage de vocabulaires reconnus internationalement (comme FOAF pour les personnes ou Dublin Core pour les documents), les données restent isolées et difficilement réutilisables par d'autres établissements.

Dès lors, la question centrale de ce projet est la suivante :

Comment concevoir une modélisation sémantique unifiée (RDFS/OWL) des ressources de l'enseignement supérieur qui permette non seulement de structurer les connaissances et d'assurer l'interopérabilité entre les systèmes, mais aussi de garantir la cohérence logique des données grâce au raisonnement automatique ?

3. Travail Réalisé

3.1 Choix de Modélisation (RDFS)

Pour répondre aux besoins d'interopérabilité et de raisonnement, nous avons structuré notre modèle autour de trois axes principaux : la hiérarchie des classes, les propriétés et l'alignement avec les standards.

- a. **Hiérarchie des Classes (Taxonomie)** : Nous avons organisé les concepts clés en une hiérarchie stricte pour permettre l'héritage des propriétés

- Personne comme classe racine pour tous les acteurs humains.
 - Les sous-classes Enseignant, Etudiant, et PersonnelAdministratif héritent des attributs communs (nom, email).
 - La classe Doctorant est défini comme sous-classe d'Etudiant (car il suit une formation) tout en ayant des spécificités de recherche.
 - La classe EnseignantChercheur hérite à la fois d'Enseignant et de Chercheur (héritage multiple), ce qui permet de lui attribuer des charges d'enseignement et des activités de recherche.
- Formation et Structure : Une distinction claire est faite entre les structures administratives (Université, Faculté, Département, Laboratoire) et les contenus pédagogiques (Formation, UnitéEnseignement).

- b. **Relations et Propriétés** : Les relations entre les entités sont modélisées par des ObjectProperties avec des domaines et des co-domaines (Range) précis pour contraindre les liens valides :

- ens:inscritDans relie obligatoirement un Etudiant (Domaine) à une Formation (Range).
- ens:dispense relie un Enseignant à une UnitéEnseignement.
- ens:membreDe connecte un Chercheur à un Laboratoire.

c. **Alignement avec les Standards (Vocabulaires Externes)** : Pour garantir l'interopérabilité, nous ne réinventons pas la roue. Nous avons aligné nos concepts avec les vocabulaires standards du Web Sémantique :

- FOAF (Friend of a Friend) : Notre classe `ens:Personne` est une sous-classe de `foaf:Person`. Les propriétés `ens:nom` et `ens:email` sont respectivement des sous-propriétés de `foaf:name` et `foaf:mbox`.
- Dublin Core (DC) : Utilisé pour les métadonnées des publications et des formations (`dc:title`, `dc:date`), facilitant ainsi le référencement bibliographique standardisé.

3.2 Contraintes OWL Utilisées

L'ontologie `ens-sup.owl` enrichit le schéma RDFS en introduisant des contraintes logiques permettant de formaliser les règles métier du domaine universitaire. Contrairement à RDFS, qui permet uniquement de définir des classes, des hiérarchies et des propriétés, OWL offre la possibilité d'exprimer des restrictions, des cardinalités minimales, des relations de disjonction ainsi que des alignements avec des vocabulaires standards du Web sémantique. Ces contraintes garantissent la cohérence des données et permettent l'inférence automatique de nouvelles connaissances.

a. Restrictions existentielles et cardinalités minimales

Une première contrainte impose qu'un étudiant soit inscrit dans au moins une formation. Cette règle est formalisée par une restriction de cardinalité minimale sur la propriété `inscritDans`. En logique de description, cela s'écrit :

$\text{Etudiant} \sqsubseteq (\geq 1 \text{ inscritDans})$

Cela signifie que tout individu déclaré comme Étudiant doit obligatoirement être lié, via la propriété `inscritDans`, à au moins une instance de la classe Formation.

De la même manière, une unité d'enseignement doit être enseignée par au moins un enseignant. Cette contrainte est exprimée par :

$\text{UnitéEnseignement} \sqsubseteq (\geq 1 \text{ estEnseigneePar})$

Ainsi, aucune unité d'enseignement ne peut exister sans être associée à au moins un enseignant.

Par ailleurs, une formation doit comporter au moins une unité d'enseignement :

$\text{Formation} \sqsubseteq (\geq 1 \text{ contientUE})$

Ces contraintes assurent qu'aucune formation ni aucune unité d'enseignement ne soient définies de manière incomplète.

b. Cardinalités et cohérence structurelle

L'ontologie impose également certaines contraintes structurelles afin d'éviter les incohérences dans les données. L'utilisation de restrictions de cardinalité minimale garantit qu'un étudiant,

une formation ou une unité d'enseignement respecte les règles organisationnelles du système universitaire.

Ces contraintes permettent au moteur d'inférence de détecter automatiquement des incohérences si, par exemple, un étudiant n'est relié à aucune formation ou si une formation ne contient aucune unité d'enseignement.

c. Classe définie par restriction

La classe *EnseignantChercheur* est définie comme une sous-classe d'*Enseignant* et est soumise à une restriction sur la propriété *membreDe*, indiquant qu'un enseignant-chercheur appartient à au moins un laboratoire. En logique de description :

$\text{EnseignantChercheur} \sqsubseteq \text{Enseignant}$
 $\text{EnseignantChercheur} \sqsubseteq \exists \text{membreDe}.\text{Laboratoire}$

Cela signifie qu'un individu de type *EnseignantChercheur* est nécessairement un enseignant impliqué dans une activité de recherche au sein d'un laboratoire.

d. Classes disjointes

Enfin, l'ontologie déclare que certaines classes sont disjointes :

$\text{Etudiant} \sqcap \text{Enseignant} \sqsubseteq \perp$

$\text{Etudiant} \sqcap \text{PersonnelAdministratif} \sqsubseteq \perp$

$\text{Enseignant} \sqcap \text{PersonnelAdministratif} \sqsubseteq \perp$

Ces contraintes signifient qu'un individu ne peut pas appartenir simultanément à ces catégories. Si une telle situation se produit dans les données, un moteur d'inférence détectera automatiquement une incohérence logique.

Cette déclaration de disjonction contribue à garantir la cohérence globale du modèle sémantique.

e. Alignement avec des vocabulaires standards

L'ontologie s'aligne également avec des vocabulaires du Web sémantique afin d'améliorer l'interopérabilité des données. La classe *Personne* est déclarée comme sous-classe de *foaf:Person*. Les propriétés *nom* et *email* sont respectivement alignées avec *foaf:name* et *foaf:mbox*, tandis que *titre* et *dateCreation* sont alignées avec *dc:title* et *dc:date*.

3.3 Exemples d'Inférences

L'un des points forts de notre modélisation est la capacité du système à déduire de nouvelles informations non présentes explicitement dans les données brutes.

a. Analyse du Graphe de Données

Le graphe de données ci-dessous représente les relations sémantiques entre les différentes entités modélisées (Enseignants, Étudiants, Formations, etc.).

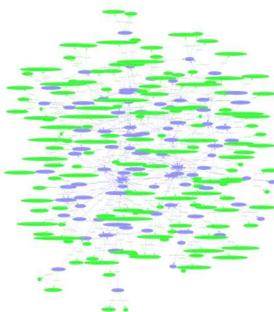


Figure 1: Graphe de données global

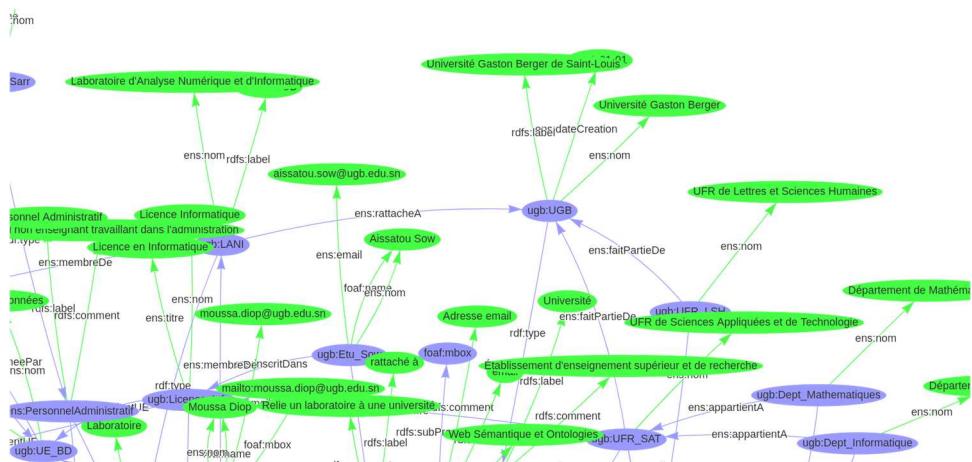


Figure 2: Graphe de données (vue zoomée)

Le graphe permet de visualiser la connectivité des données instanciées dans les fichiers Turtle (UGB). On y observe les nœuds représentant les individus (ex: Prof_Diop, Etu_Fall) et les arcs représentant les propriétés (ex: enseigne, inscritDans).

b. Inférence de Types (RDFS)

Grâce à la hiérarchie 'rdfs:subClassOf', toute instance d'Étudiant ou d'Enseignant est automatiquement inférée comme étant une 'Personne'. Cela permet d'interroger globalement l'annuaire via la classe racine 'Personne'.

- Exécution des Requêtes et Analyse des Résultats

Requête 1 : Lister les enseignants et leurs UE

Cette requête récupère les enseignants et les unités d'enseignement qu'ils dispensent.

```
# REQUÊTE 1 : Lister tous les enseignants et les UE qu'ils enseignent

SELECT ?enseignant ?nomEnseignant ?ue ?nomUE
WHERE {
    ?enseignant a ens:Enseignant .
    ?enseignant ens:nom ?nomEnseignant .
    ?enseignant ens:enseigne ?ue .
    ?ue ens:nom ?nomUE .
}
ORDER BY ?nomEnseignant
```

Requête SPARQL 1

Résultat (Sans Inférence) :

Result

1	-----			
2	enseignant	nomEnseignant	ue	
3	=====			
4	<http://www.ugb.sn/ressources/UGB#Prof_Ndiaye>	"Fatou Ndiaye"	<http://www.ugb.sn/ressources/UGB#UE_BD>	"Bases de Données"
5	<http://www.ugb.sn/ressources/UGB#Prof_Ba>	"Ibrahima Ba"	<http://www.ugb.sn/ressources/UGB#UE_Algo>	"Algorithmique"
6	<http://www.ugb.sn/ressources/UGB#Prof_Diop>	"Moussa Diop"	<http://www.ugb.sn/ressources/UGB#UE_WebSem>	"Web Sémantique"
7	-----			
8				

Analyse : Résultat obtenu sans inférence particulière. Les relations sont explicites dans les données.

Requête 2 : Étudiants par Formation

Affichage des étudiants inscrits dans chaque formation.

```
# REQUÊTE 2 : Trouver les étudiants inscrits dans chaque formation

SELECT ?etudiant ?nomEtu ?formation ?nomFormation
WHERE {
    ?etudiant a ens:Etudiant .
    ?etudiant ens:nom ?nomEtu .
    ?etudiant ens:inscritDans ?formation
    ?formation ens:nom ?nomFormation
}
ORDER BY ?nomFormation
```

Requête SPARQL 2

Résultat (Sans Inférence) :

Result

1	2 etudiant	3 nomEtu	4 formation	5 nomFormation	6	7
2	<http://www.ugb.sn/ressources/UGB#Etu_Sow>	"Aissatou Sow"	<http://www.ugb.sn/ressources/UGB#Licence_Info>	"Licence Informatique"		
3	<http://www.ugb.sn/ressources/UGB#Etu_Fall>	"Amadou Fall"	<http://www.ugb.sn/ressources/UGB#Master_GDIL>	"Master GDIL"		
4						
5						
6						
7						

Analyse : Liste des étudiants liés aux formations via la propriété 'ens:inscritDans'.

Requête 4 : Doctorants et Directeurs de Thèse

Liste des doctorants, directeurs et écoles doctorales.



```
# REQUÊTE 4 : Trouver les doctorants et leurs directeurs de thèse

SELECT
?doctorant ?nomDoc ?directeur ?nomDir ?ecoleDoctorale ?nomE
WHERE {
?doctorant a ens:Doctorant .
?doctorant ens:nom ?nomDoc .
?doctorant ens:dirigePar ?directeur
?directeur ens:nom ?nomDir .
?doctorant ens:inscritDansE ?ecoleDoctorale
?ecoleDoctoraleDens:nom ?nomED .
}
```

Requête SPARQL 4

Résultat (Sans Inférence) :

Result

1	2 doctorant	3 nomDoc	4 directeur	5 nomDir	6 ecoleDoctorale
2	nomED				
3					
4	<http://www.ugb.sn/ressources/UGB#Doc_Gueye>	"Ousmane Gueye"	<http://www.ugb.sn/ressources/UGB#Prof_Diop>	"Moussa Diop"	
5	<http://www.ugb.sn/ressources/UGB#ED_SEV>	"École Doctorale Sciences et Technologies"			
6					

Analyse : Relation directe exploitant les propriétés 'ens:dirigePar' et 'ens:inscritDansED'.

b. Classification Automatique (Enseignant-Chercheur)

Problème : Dans les données brutes, un individu comme 'Prof_Ndiaye' est déclaré comme 'Enseignant' et possède une relation 'membreDe' vers un 'Laboratoire', mais n'est pas explicitement typé 'EnseignantChercheur'.

Solution OWL : Nous avons défini la classe 'EnseignantChercheur' comme une classe équivalente à l'intersection de la classe 'Enseignant' et de la restriction 'membreDe some Laboratoire'.

Résultat : Le raisonneur infère automatiquement que 'Prof_Ndiaye' est un 'EnseignantChercheur'. Sans cette inférence, une requête SPARQL sur les enseignants-chercheurs ne rentrait aucun résultat.

- Exécution des Requêtes et Analyse des Résultats

Requête 3 : Recherche des Enseignants-Chercheurs

Identification des individus de type EnseignantChercheur.

```
# REQUÊTE 3 : Trouver les EnseignantChercheurs déclarés EXPLICITEMENT

SELECT ?ec ?nomEC
WHERE {
    ?ec a ens:EnseignantChercheur .
    ?ec ens:nom ?nomEC .
}
```

Requête SPARQL 3

Résultat (Sans Inférence) :

Result

```
1 -----
2 | ec | nomEC |
3 =====
4 -----
5 -----
```

Résultat (Avec Inférence) :

	num	?ec	?nomEC
1		<http://www.ugb.sn/ressources/UGB#Prof_Diop>	Moussa Diop
2		<http://www.ugb.sn/ressources/UGB#Prof_Ndiaye>	Fatou Ndiaye

Analyse :

- Sans inférence : Aucun résultat. Dans le fichier Turtle, personne n'est explicitement déclaré comme 'EnseignantChercheur'.
- Avec inférence (OWLRL) : Le raisonneur déduit que les individus qui sont à la fois 'Enseignant' et membres d'un 'Laboratoire' (intersection de classes) sont des 'EnseignantChercheur'. Par exemple, Moussa Diop est classé automatiquement.

Requête 5 : Lister toutes les Personnes

Récupération de toutes les instances de la classe Personne.

```
# REQUÊTE 5 : Lister toutes les Personnes
SELECT ?personne ?nomPersonne
WHERE {
    ?personne a ens:Personne .
    ?personne ens:nom ?nomPersonne
}
```

Requête SPARQL 5

Résultat (Sans Inférence) :

Result

```
1 -----
2 | personne | nomPersonne |
3 =====
4 -----
5 -----
```

Résultat (Avec Inférence) :

	num	?personne	?nomPersonne
1		<http://www.ugb.sn/ressources/UGB#Prof_Diop>	Moussa Diop
2		<http://www.ugb.sn/ressources/UGB#Prof_Ndiaye>	Fatou Ndiaye
3		<http://www.ugb.sn/ressources/UGB#Etu_Diop>	Abdoulaye Diop
4		<http://www.ugb.sn/ressources/UGB#Etu_Fall>	Arnaudou Fall
5		<http://www.ugb.sn/ressources/UGB#Etu_Sow>	Aissatou Sow
6		<http://www.ugb.sn/ressources/UGB#Doc_Gueye>	Ousmane Gueye
7		<http://www.ugb.sn/ressources/UGB#Admin_Sarr>	Mariama Sarr

Analyse :

- Sans inférence : Aucun résultat (ou partiel) si la classe 'Personne' n'est pas instanciée directement.
- Avec inférence : Le raisonneur utilise la hiérarchie des classes (rdfs:subClassOf). Puisque 'Enseignant', 'Etudiant', etc. sont des sous-classes de 'Personne', toutes leurs instances sont inférées comme étant des 'Personne'.

3.4 Classification Automatique (Enseignant-Chercheur)

Problème : Dans les données brutes, un individu comme 'Prof_Ndiaye' est déclaré comme 'Enseignant' et possède une relation 'membreDe' vers un 'Laboratoire', mais n'est pas explicitement typé 'EnseignantChercheur'.

Solution OWL : Nous avons défini la classe 'EnseignantChercheur' comme une classe équivalente à l'intersection de la classe 'Enseignant' et de la restriction 'membreDe some Laboratoire'.

Résultat : Le raisonneur infère automatiquement que 'Prof_Ndiaye' est un 'EnseignantChercheur'. Sans cette inférence, une requête SPARQL sur les enseignants-chercheurs ne retournerait aucun résultat.

4. Perspectives

Ce travail pose les bases d'un système d'information sémantique universitaire. Les perspectives d'évolution sont nombreuses :

- Interconnexion (LOD) : Lier nos données avec celles d'autres universités ou référentiels (DBpedia, Wikidata) pour enrichir les profils chercheurs.
- Moteur de Recherche Facetté : Créer une interface permettant de filtrer les ressources par facettes (par UFR, par domaine de recherche, par type de formation).
- Système de Recommandation : Suggérer des UE optionnelles aux étudiants en fonction de leur parcours ou des projets de recherche liés.