

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Chadli Bendjedid

Faculté des Sciences et de la Technologie

Département d'Informatique



الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة الشاذلي بن جديد

كلية العلوم والتكنولوجيا

قسم الآلة

MEMOIRE

Présentée par

Touati Zineb

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Vers un système d'apprentissage basé linked data

Soutenue le : / / 2020

Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Président	Mme. Ferdenache Ahlam	MCB	Chadli Bendjedid El-Tarf
Rapporteur	Mr. Chemam Chaouki	MAA	Chadli Bendjedid El-Tarf
Examineur	Mme. Zekri Meriem	MCB	Chadli Bendjedid El-Tarf

Année Universitaire : 2019/2020

Remerciements

Tout d'abord, nous remercions Dieu qui nous a bénis avec la grâce de l'étude et

Sommes heureux de terminer notre projet pour la fin de l'étude.

En témoignage de notre gratitude et de notre profond respect nous adressons

Ensuite nos vifs et sincères remerciements à :

A notre encadreur, Dr. CHEMAM Chaouki pour sa disponibilité, ses conseils et

Son encouragement.

A nos amis qui nous ont apporté leurs aides à l'élaboration de ce mémoire, nous

Leurs devons beaucoup.

Les membres du jury pour leurs efforts et leurs soins apportés à notre travail.

Aux enseignants de notre université et du département d'informatique.

Nos familles pour leurs soutiens et leurs encouragements au long de toutes ces

Années.

*À ma famille, qui a toujours cru en moi et en mes capacités, qui a toujours existé
pour moi*

*Stimuler et soutenir moi et m'aider à traverser les moments difficiles et
surmonter mes
Problèmes.*

*Pour mon père ma vie Ramdane, ma mère Farida est mon oeil
Mon cher mari Salah eddine " Mon compagnon, sans vos encouragements
continus, je n'aurais pas atteint ce succès "*

Ma chère soeur: "Noussa"

A mes cher frères Ala , imad , Mohamed Lamine

*À ma deuxième famille, qui m'a procuré une atmosphère d'étude, qui m'a
encouragé à réussir et à terminer mon travail*

*Il n'y a pas de dévotion qui puisse exprimer mes sentiments sincères, pour leur
patience*

*Illimitée, Le contenu de l'encouragement, aidez-les, dans le témoignage de mon
profond*

Amour et respect De grands sacrifices.

Pour mes tantes

Pour mes oncles

Mes amies

*Pour leur grand amour et leur soutien, ils le trouvent ici pour exprimer ma
grande gratitude.*

AUX professeurs de département « Informatique »

À le professeur le plus merveilleux : Dr. Chemam Chaouki

Et

Mes chers amis Et à toute ma famille et à tous ceux que j'aime.

Table des matières

Remerciements	2
Dédicace	3
Table des matières	4
Liste des figures	7
Liste des acronymes	9
Introduction Générale.....	11
1. Contexte du projet et problématique	11
2. Motivations.....	11
3. Objectifs	11
4. Contenu du mémoire	12
Chapitre 1 : Généralité sur Web Sémantique	13
1. Introduction :.....	13
2. Web des données (Linked data):.....	15
3. Langages du web sémantique:	18
4. Langages OWL:	23
5. Ontologies:.....	25
6. SPARQL (Protocol and RDF Query Langage):.....	28
Conclusion:.....	31
Chapitre 2 : Analyse et conception d'un système d'apprentissage basé sur linked data	32
1. Etape1 : Analyse des besoins	32
1.1. Besoins fonctionnelles :.....	32
1.2. Besoins non fonctionnelles :.....	32
2. Spécifications fonctionnelles :	33
2.1. Diagramme de cas d'utilisation :.....	33
2.2. Diagramme d'activité :.....	37
2.3. Conception détaillée :	38

2.3.1.	Diagramme de classe :	39
2.3.2.	Diagramme de séquence :	43
3.	L'architecteur d'application :	48
4.	Conception de la base de données :	48
5.	Etape 2 : l'ontologie et l'annotation sémantique	50
5.1.	L'ontologie :	50
5.1.1.	Les étapes de création d'ontologie :	50
5.2.	Manipulation d'ontologie :	54
5.3.	Passage de la langue naturelle à une requête SPARQL:	55
5.4.	L'annotation sémantique :	57
5.4.1.	Histoire :	58
5.4.2.	Application :	58
5.4.3.	Fonctionnement :	59
5.4.4.	Méthode standard :	59
5.4.5.	Méthode désambigüisation:	59
5.4.6.	Standards de liens:	60
5.4.7.	Problème de l'adaptation linguistique:	60
5.4.8.	Types d'annotation sémantique :	60
5.4.8.1.	Annotation vocale :	60
5.4.8.2.	Annotation image :	62
5.4.8.3.	Annotation vidéo :	62
6.	Conclusion	63
Chapitre 3 : Implémentation et Réalisation de l'application		64
1.	Environnement de développement:	64
1.1.	L'environnement Hardware :	64
1.2.	L'environnement Software :	64
2.	Présentation et Test de l'application :	69
3.	Conclusion:	83
Conclusion et Perspectives		84

Liste des figures

Figure 1 : Représenté la structure de web sémantique.....	14
Figure 2 : Architecture de Web Sémantique.....	14
Figure 3 : Représenté la relation entre URL et URN.....	16
Figure 4 : Représenté l'évolution de DBpedia 2007-2010.....	18
Figure 5 : Exemple de document XLM représenté sous forme textuelle.....	19
Figure 6 : Exemple de graphe RDF.....	20
Figure 7 : Exemple schéma RDF.....	23
Figure 8 : Représenté de trois niveaux OWL (OWL1 Lite \subset OWL1 DL \subset OWL1 Full).....	24
Figure 9 : Définition d'une ontologie représentée.....	26
Figure 10 : Architecture du mécanisme de requête SPARQL pour les référentiels RDF distribués.....	29
Figure 11 : Structure de la requête SPARQL.....	30
Figure 12 : Représentation le processus de requête SPARQL	30
Figure 13 : diagramme de cas d'utilisation (gestion des administrateurs).....	34
Figure 14 : diagramme de cas d'utilisation (gestion des enseignants).....	35
Figure 15 : diagramme de cas d'utilisation (gestion des apprenants).....	36
Figure 16 : diagramme de cas d'utilisation (gestion des visiteurs).....	37
Figure 17 : diagramme d'activité (gestion des apprenants, des enseignants et des administrateurs (système)).....	38
Figure 18 : Diagramme de classe (gestion d'administrateur).....	40
Figure 19 : Diagramme de classe (gestion d'enseignant).....	41
Figure 20 : Diagramme de classe (gestion d'apprenant).....	42
Figure 21 : Diagramme de classe (gestion de visiteur).....	43
Figure 22 : diagramme de séquence (gestion des apprenants).....	45
Figure 23 : diagramme de séquence (gestion des enseignants).....	47
Figure 24: Architecture d'application.....	48
Figure 25 : Gestion de contenu.....	49
Figure 26 : Représenté l'écran de protégé.....	51

Figure 27 : Représenté les classes d'ontologie.....	52
Figure 28 : Représenté les propriétés d'ontologie.....	52
Figure 29 : Représenté une partie des sous-classes.....	53
Figure 30 : Représenté le graphe d'ontologie.....	54
Figure 31 : Cadre pour Q / A RDF basé sur un modèle.	56
Figure 32: Architecture de l'annotation sémantique.....	58
Figure 33 : Page d'accueil de la plate-forme LINKED DATA.....	70
Figure 34 : Page de Contacter.....	71
Figure 35 : Page de Consultation des articles.....	72
Figure 36 : Page de Connexion d'enseignant.....	73
Figure 37 : Page d'accueil espace d'enseignant.....	74
Figure 38 : Page de système d'annotation (espace d'enseignant).....	75
Figure 39 : Page de Connexion d'apprenant.....	76
Figure 40: Page d'accueil d'apprenant.....	77
Figure 41 : Page de système d'annotation (espace d'apprenant).....	78
Figure 42 : Page de système d'annotation (création d'annotation).....	79
Figure 43 : Page de système d'annotation Spotlight (création d'annotation).....	79
Figure 44 : Page de Quizz (login de l'apprenant).....	80
Figure 45 : Page de Quizz de l'apprenant.....	80
Figure 46 : Page Connexion d'administrateur.....	81
Figure 47 : Page de registre d'administrateur.....	81
Figure 48 : Page accueil d'administrateur.....	82
Figure 49 : Page affichage des annotations d'administrateur.....	82

Liste des acronymes

Les abréviations utilisées dans ce manuscrit sont présentées dans le but de connaître leur signification en anglais et (ou) l'équivalent en français en cas de besoin :

HTTP/HTTPS	HyperText Transfer Protocol/ HyperText Transfer Protocol Secure
HTML	HyperText Markup Language
CSS	Cascading StyleSheet
PHP	Personal Home Page Hypertext Preprocessor
SGBD	Système de Gestion de Bases de Données
UML	Unified Modeling Language .
URI	Uniform Resource Identifier
UDI	Unified Display Interface
REC	Uniform Resource Locator
URN	Uniform Resource Locator
MCD	Modèle Conceptuel des Données
MLD	Modèle Logique des Données
WWW	World Wide Web
W3C	World Wide Web Consortium
SPRQL	Short for Simple Protocol and RDF Query Language
RDF	Resource Description Framework
RDFS	RDF Schema
N3	Notation 3
ISBN	International Standard Book Number
SSL	Secure Socket Layer
TLS	Transport Layer Security
OWL	Web Ontology Language
XMI	XML Meta-data Inter change
XSD	XML Schema Definition
SKOS	Simple Knowledge Organization System
FOAF	Friend Of A Friend
GED	Gestion Électronique des Documents
SQL	Structured Query Language
CSV	Comma-Separated Values
JSON	JavaScript Object Notation
REC	Récupération de données

L'IETF	Internet Engineering Task Force
FRC	Les requests for comments
DAML	Digital Asset Modeling Language

Introduction Générale

Avec le développement fulgurant, en termes de quantité et disponibilité, des ressources éducatives libres qui sont de plus en plus offertes gratuitement par de prestigieuses universités et fondations à travers le monde, la communauté des chercheurs en sciences de l'éducation s'est vite intéressée à leur potentiel tant pour les apprenants que pour les enseignants. En effet, un nouveau concept appelé «Linked Data » a été introduit comme une nouvelle technologie prometteuse. Ce concept pourrait faire l'objet d'une interconnexion et enrichissement des ressources éducatives en s'appuyant sur quatre principes à savoir :

- (i) utiliser des identifiants uniques.
- (ii) utiliser des URI HTTP.
- (iii) fournir des informations à l'aide de RDF et SPAQL.
- (iv) créer des liens vers d'autres ressources.

Cependant, ces potentialités offertes par le Linked Data n'ont pas encore pleinement exploitées.

1. Contexte du projet et problématique

De ce fait, dans cette mémoire nous nous sommes intéressés à développer un système pour l'apprentissage automatique en ligne qui se base principalement sur les principes de Linked Data en tenant compte de l'aspect pédagogique.

Comment convertir une question de langage naturel a une requête automatisé par l'annotation d'un document pédagogie?

2. Motivations

Avec la quantité croissante de données sémantiques disponibles sur le Web, il existe un besoin urgent de systèmes permettant aux apprenants d'accéder à ce corpus de connaissances. Une attention particulière a été accordée aux systèmes de réponse aux questions, car ils permettent aux apprenants d'exprimer arbitrairement des besoins d'information complexes d'une manière simple et intuitive. Afin que les apprenants, lors de leur processus d'apprentissage, puissent enrichir leurs contenus pédagogiques par des données fournies par le Web de données.

3. Objectifs

L'objectif de ce mémoire est de mettre en place un système intelligent pour aider l'apprenant dans les cours en ligne et de le faire en utilisant la technologie d'annotation sur un document

pédagogique, et de faciliter la communication avec leurs enseignants afin de passer une question en langage naturel via l'application. Et cela utilise Linked Data.

4. Contenu du mémoire

Ce mémoire, se divise en trois chapitres principaux :

- Le premier chapitre présentera généralité sur le web sémantique.
- Le deuxième chapitre décrira l'analyse et conception de système.
- Le dernier chapitre est dédié implémentation et réalisation d'application

Finalement, nous terminons par une conclusion générale et quelques perspectives intéressantes.

Chapitre 1 : Généralité sur Web Sémantique

Résumé : L'intérêt croissant porté à la recherche d'informations sur le Web a donné lieu à l'initiative du Web sémantique. L'objectif est de rendre les ressources du web facilement accessibles non seulement êtres humains mais et surtout à la machine grâce à la représentation sémantique de leur contenu. Pour rendre possible cette représentation, il est important de mettre l'accent sur des langages déstructuration sémantique des connaissances. Celle-ci permettra également de fournir le vocabulaire et la structure des métadonnées associées aux différentes ressources

1. Introduction :

Le Web sémantique désigne un ensemble de technologies visant à rendre le contenu des ressources du World Wide Web accessible et utilisable par les programmes et agents logiciels grâce à un système de métadonnées formelles, utilisant notamment la famille de langages développés par le W3C. Comme l'écrit en substance Tim Berners-Lee, « le Web sémantique est ce que nous obtiendrons si nous réalisons le même processus de globalisation sur la représentation des connaissances que celui que le Web fit initialement sur l'hypertexte »[1]. Partant de ce fait, le Web sémantique peut se définir comme la mise en relation de l'information pour permettre aux ordinateurs de comprendre la sémantique (le sens) de l'information, Il s'appuie sur les standards du Web, tels que HTTP et URI mais plutôt qu'utiliser ces standards uniquement pour faciliter la navigation par les êtres humains, le Web des données les étend pour partager l'information également entre machines. Cela permet d'interroger automatiquement les données, quels que soient leurs lieux de stockage, et sans avoir à les dupliquer. Tim Berners-Lee, directeur du W3C, a inventé et défini le terme linked data (Les données liées) et son synonyme Web of data au sein d'un ouvrage portant sur l'avenir du Web sémantique. Les données liées consistent à utiliser le Web pour connecter des données liées qui n'étaient pas précédemment liées, ou à utiliser le Web pour réduire les obstacles à la liaison de données actuellement liées à l'aide d'autres méthodes. Plus précisément, Wikipedia définit les données liées comme «un terme utilisé pour décrire une meilleure pratique recommandée pour exposer, partager et connecter des éléments de données, d'informations et de connaissances sur le Web sémantique à l'aide d'URI et de RDF ».Ce site existe pour fournir un foyer ou des pointeurs vers des ressources provenant de la communauté des données liées [2].

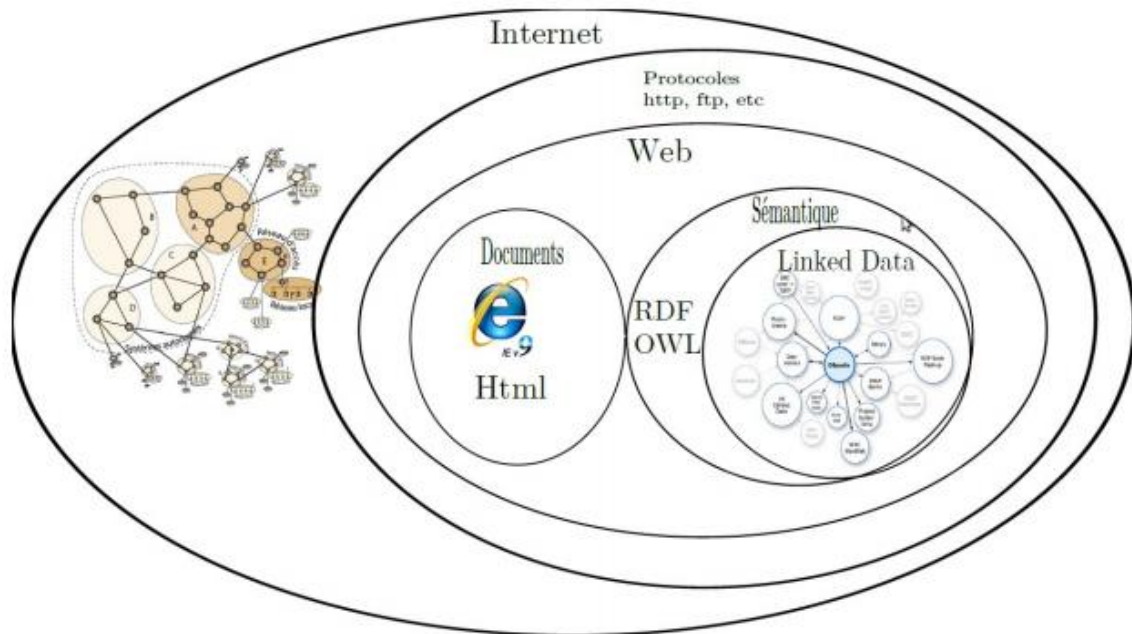


Figure 1 : Représenté la structure de web sémantique [6].

La différence philosophique majeure entre ces deux technologies est que le web actuel est un ensemble de documents basés essentiellement sur l'identifiant URI du protocole HTTP et sur le langage HTML, dont la recherche se fait à travers des mots clés et utilisables par l'homme tandis que le web sémantique est un ensemble de connaissances basées sur le document XML (eXtensible Markup Language) et RDF (Resource Description Framework). D'après Tim Berners-Lee [3], aujourd'hui, les humains sont les seuls êtres à avoir la capacité de comprendre ce qu'ils trouvent et les seuls à pouvoir décider de ce qu'ils veulent chercher réellement. Cependant, par quels moyens les moteurs de recherche nous aident-ils? En d'autres termes, comment les ressources textuelles ou multimédias doivent elles être sémantiquement étiquetées par des métadonnées afin que les agents logiciels puissent les exploiter [4] ? La représentation explicite des contenus des ressources documentaires du Web est rendue possible grâce aux ontologies. Ces ontologies représentent une technologie clef pour la mise en œuvre de ce Web Sémantique.

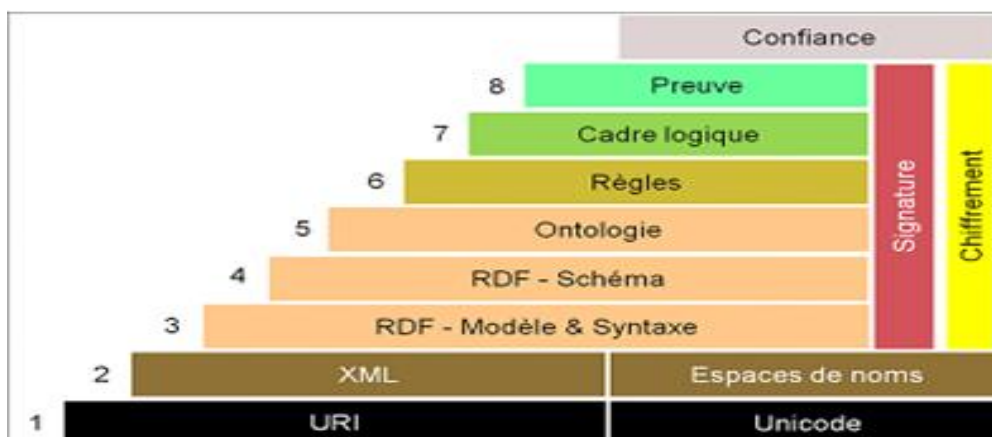


Figure 2 : Architecture de Web Sémantique [5].

2. Web des données (Linked data):

Les données liées sont des principaux piliers du Web sémantique , également connu sous le nom de Web of Data. Le Web sémantique consiste à créer des liens entre des ensembles de données qui sont compréhensibles non seulement pour les humains, mais aussi pour les machines, et les données liées fournissent les meilleures pratiques pour rendre ces liens possibles. En d'autres termes, Linked Data est un ensemble de principes de conception pour le partage de données interconnectées lisibles par machine sur le Web.

2.1.les règles de jeu de données liées :

Plus il y a de choses, d'événements, de personnes, de lieux, etc. connectés entre eux, plus le Web of Data est puissant. Cependant, afin de lier, fusionner et intégrer d'énormes ensembles de données provenant de sources disparates, certaines directives de base doivent être suivies. L'inventeur du World Wide Web et le créateur et défenseur du Web sémantique et des données liées, Sir Tim Berners-Lee, ont défini les quatre principes de conception des données liées dès 2006.

2.2.Composants :

a. Uniforme Resource Identifier URI :

Un URI, de l'anglais Uniform Resource Identifier, soit littéralement identifiant uniforme de ressource, est une courte chaîne de caractères identifiant une ressource sur un réseau (par exemple une ressource Web) physique ou abstraite, et dont la syntaxe respecte une norme d'Internet mise en place pour le World Wide Web .

b. Principe :

Un URI doit permettre d'identifier une ressource de manière permanente, même si la ressource est déplacée ou supprimée.

c. Relation entre URL et URN :

Un URI peut être de type « locator » ou « name » ou les deux.

- Un Uniform Resource Locator (URL) est un URI qui, outre le fait qu'il identifie une ressource sur un réseau, fournit les moyens d'agir sur une ressource ou d'obtenir une représentation de la ressource en décrivant son mode d'accès primaire ou « emplacement » réseau. Par exemple, l'URL <http://www.wikipedia.org/> est un URI qui identifie une ressource (page d'accueil Wikipédia) et implique qu'une représentation de cette ressource (une page HTML en caractères encodés) peut être obtenue via le protocole HTTP depuis un réseau hôte appelé www.wikipedia.org.

- Un Uniform Resource Name (URN) est un URI qui identifie une ressource par son nom dans un espace de noms. Un URN peut être employé pour parler d'une ressource sans que cela préjuge de son emplacement ou de la manière de la référencer. Par exemple, l'URN `urn:isbn:0-395-36341-1` est un URI qui, étant un numéro de l'International Standard Book Number (ISBN), permet de faire référence à un livre, mais ne suggère ni où, ni comment en obtenir une copie réelle.

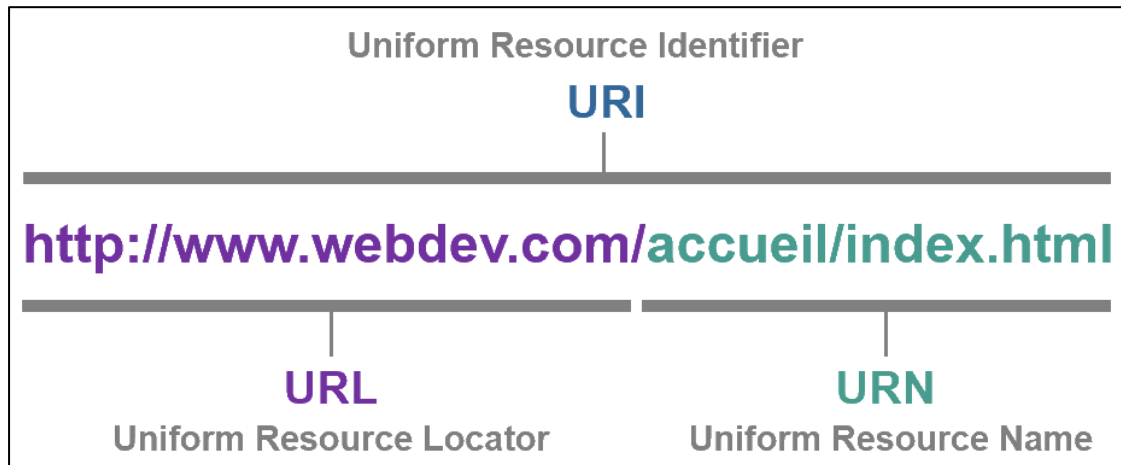


Figure 3 : Représenté la relation entre URL et URN.

2.3. Données ouvertes liées (Linked open data) :

a. Définition des données ouvertes liées:

Le Linked Open Data ou «données ouvertes liées» consiste à rendre libre de toute licence des bases de données préalablement liées entre elles selon le modèle du Linked Data de Tim Berners-Lee. Ce système rend la recherche sur le web plus efficace et dynamique pour l'utilisateur en reliant d'autres sources d'informations pertinentes et insoupçonnées à la recherche initiale. Le fait que les données soient libres permet leur accès et leur réutilisation sans aucune restriction.

Le Linked Open Data s'inscrit dans le web 3.0 qui est un web de données. À la différence du Web 1.0 qui regroupe quantité de documents individuels sous la forme de silos de données isolés des uns des autres, le web 3.0 est constitué de données brutes respectant toutes les mêmes standards du Web, ce qui rend leur utilisation, leur partage et leur interopérabilité très facile. Le Web des données prend forme à travers un modèle proposé par le W3C, le Web sémantique qui est un modèle commun permettant le partage et la réutilisation de données sont surtout présente par l'entremise d'une application, d'une compagnie ou d'une communauté. Ainsi, le Web des données offre une immense quantité de données disponibles sur le Web dans un format standard afin que cette masse d'informations puisse être traitée par des outils du web sémantique.

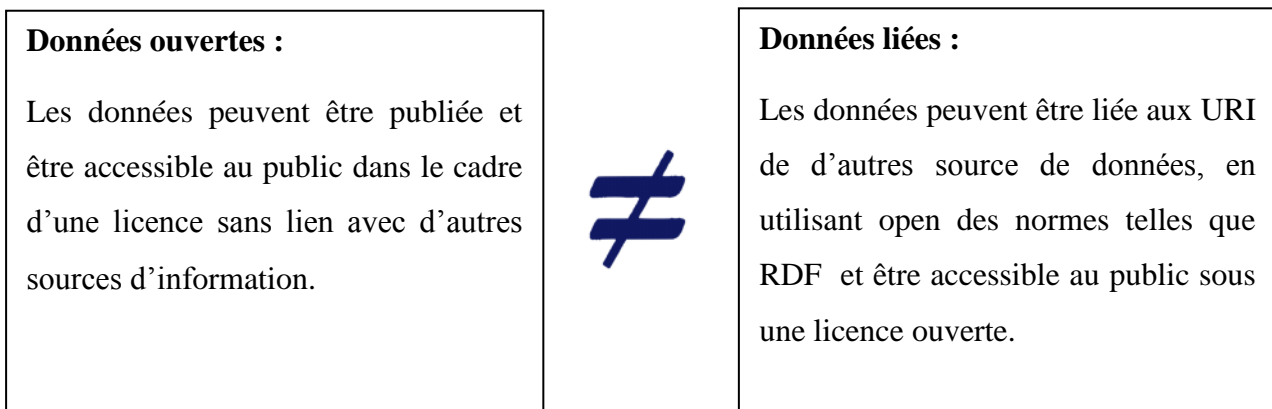
L'accessibilité doit être garantie autant aux données qu'aux liens entre celles-ci afin de créer une vaste toile d'information et de connaissances.

b. Principe de Technique :

Le Linked Open Data repose sur 4 principes de base essentiels :

- i. Les données doivent être libres de toute licence pour que des liens soient facilement faits entre elles.
- ii. Chaque lien utilise le modèle Ressource Description Framework (RDF). Un RDF triplet est une association triple de liens soi: sujet-prédicat-objet. Ce modèle permet de décrire formellement les ressources Web et leurs métadonnées, afin d'en permettre un traitement automatique et une certaine interopérabilité).
- iii. Afin d'être localisée, chaque ressource possède une adresse unique et permanente en ligne, Uniform Ressource Identifier (URI).
- iv. Les données doivent être mises en ligne suivant le protocole standard HTTP.

c. Données liées VS données ouvertes :



2.4.DBpedia :

En un des exemples les plus connus et de grande taille, qui a adopté les normes du réseau linked open data et celles du Web sémantique, faisant qu'elle a rapidement été fortement interconnectée à d'autres dépôts du web de données tel que GeoNames, MusicBrainz, CIA World Factbook, le projet Gutenberg et Eurostat.

L'accès aux dépôts de données se fait avec des requêtes sur la base de données via SPARQL. Les informations étant stockées avec Resource Description Framework, on peut aussi récupérer des documents ressource en relation avec un concept directement via une URI, avec les formats CSV ou RDF (notamment via les formats N-Triple, N3, JSON, XML) [5].

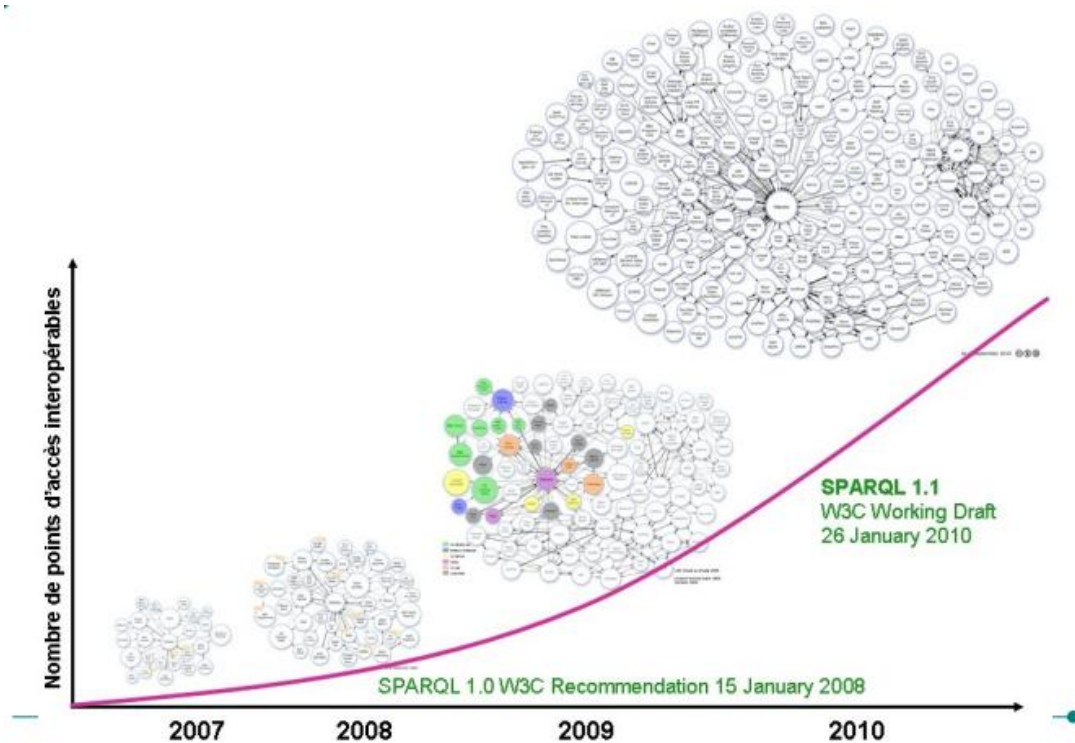


Figure 4 : Représenté l'évolution de DBpedia 2007-2010 [5].

3. Langages du web sémantique:

Le Web sémantique est une infrastructure liée à l'utilisation de connaissances formalisées, en plus du contenu informel actuel du Web. Cette infrastructure doit permettre de localiser, d'identifier et de transformer des ressources du web de manière robuste et saine tout en renforçant l'esprit d'ouverture de celui-ci, avec sa diversité d'utilisateurs. Elle doit également s'appuyer sur un certain niveau de consensus portant sur les langages de représentation ou sur les ontologies utilisées [7]. Le Web sémantique devrait pouvoir être manipulé par les machines, d'où la nécessité de disposer de langages appropriés pour :

- exprimer les données et les métadonnées.
- exprimer les ontologies.
- décrire les services [8].

A l'état actuel de la technologie, il existe trois sortes de langages :

- les langages d'assertions (RDF et cartes topiques).
- les langages de définition d'ontologies pour le Web (OWL, XML, RDF etc.).
- les langages de description et de composition de Services (UDDI et autres).

3.1. Langage XML :

Le langage de balisage extensible (eXtensible Markup Language) est un langage de balisage qui définit un ensemble de règles pour coder les documents dans un format à la fois lisible par l'homme et lisible par la machine . La spécification XML 1.0 du World Wide Web Consortium

[9] de 1998 [10] et plusieurs autres spécifications connexes [11] toutes des normes ouvertes gratuites - définissent XML [12].

Les objectifs de conception de XML mettent l'accent sur la simplicité, la généralité et la convivialité sur Internet [13]. Il s'agit d'un format de données textuelles avec un support solide via Unicode pour différentes langues humaines . Bien que la conception de XML se concentre sur les documents, le langage est largement utilisé pour la représentation de structures de données arbitraires [14] telles que celles utilisées dans les services Web .

Plusieurs systèmes de schéma existent pour faciliter la définition des langages basés sur XML, tandis que les programmeurs ont développé de nombreuses interfaces de programmation d'application (API) pour faciliter le traitement des données XML.

Le contenu d'un document Web peut être facilement traité parce que le XML fournit un moyen d'échanger des informations sur le web. Le XML permet de séparer la présentation du contenu des documents. C'est un langage de base pour le Web sémantique, conçu essentiellement pour décrire des données en se concentrant sur leur structure et aussi pour assurer leur interopérabilité. Pour rendre possible cette méthode, XML utilise un «Document Type Définition» (DTD) et un XML Schéma pour un « modèle » de données, comme illustré par (figure 5).

```
<?xml version="1.0" encoding="utf-8" ?>
<personne>
  <prenom>Donald</prenom>
  <nom>Knuth</nom>
  <date_naissance>1938-01-10</date_naissance>
  <adresse>
    <société>Stanford University</société>
    <code_postal>CA 94305</code_postal>
    <pays code = "ISO-3166">US</pays>
  </adresse>
  <profession>informaticien</profession>
</personne>
```

Figure 5 : Exemple de document XML représenté sous forme textuelle [15].

3.2.Langage RDF (Ressource Description Framework) :

RDF est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées, de façon à permettre le traitement automatique de telles descriptions. Développé par le W3C, RDF est le langage de base du Web sémantique. L'une des syntaxes de ce langage est RDF/XML. D'autres syntaxes de RDF sont apparues ensuite, cherchant à rendre la lecture plus compréhensible ; c'est le cas par exemple de Notation3 (ou N3).

En annotant des documents non structurés et en servant d'interface pour des applications et des documents structurés (par exemple bases de données et GED) RDF permet une certaine interopérabilité entre des applications échangeant de l'information non formalisée et non structurée sur le Web [5].

Le langage RDF vient donc remplir ce vide puisque c'est un langage qui permet de représenter des informations sur des ressources sous la forme de graphes orientés avec des nœuds et des arcs étiquetés (**figure 6**).



Figure 6 : Exemple de graphe RDF [16].

a. La structure RDF :

RDF est représenté par des triplets, graphes et syntaxe

- **Qu'est ce qu'un triplet ?** RDF est une syntaxe pour représenter des données sur le Web de manière générale.
- **RDF décompose chaque élément d'information en triplets:**

Sujet	Une ressource qui peut être identifiée par un URI.
Prédicat	Une spécification réutilisée et identifiée par URI de la relation.
Objet	Une ressource ou constante à laquelle le sujet est lié.

b. Vocabulaire :

La structure de RDF est extrêmement générique et sert de base à un certain nombre de schémas ou vocabulaires dédiés à des applications spécifiques. Une partie de ces vocabulaires est spécifiée par le W3C, comme les langages d'ontologie RDFS et OWL, ou le vocabulaire SKOS pour la représentation des thésaurus et autres vocabulaires structurés. D'autres vocabulaires RDF, sans être spécifiés par le W3C, sont néanmoins utilisés largement et constituent des standards de fait dans la communauté du Web Sémantique, comme FOAF qui est destiné à la représentation des personnes.

Un vocabulaire est un modèle de données comprenant des classes, propriétés et relations qui peuvent être utilisées pour décrire vos données et métadonnées.

- ✓ Vocabulaires RDF sont des ensembles de termes utilisés pour décrire les choses.
- ✓ Un terme est soit une classe, soit une propriété.
- ✓ Propriétés de type objet (les relations).
- ✓ Propriétés de type de données (attributs).

Que sont les classes, les relations et les propriétés?

- ✓ **Classe :** Une construction qui représente les choses dans le monde réel et / ou des informations, **par exemple :** (une personne, une organisation, des concepts tels que «santé» ou «liberté»).
- ✓ **Relation :** Un lien entre deux classes, comme le lien entre un document et l'organisation qui l'a publiée (**par ex.** organisation publie un document), ou le lien entre une carte et la région géographique qu'il représente (**par ex.** carte représente région géographique). En RDF, les relations RDF sont encodées comme des propriétés de type d'objet.
- ✓ **Propriété :** Une caractéristique d'une classe dans un domaine particulier, comme le nom légal d'une organisation ou à la date et l'heure de l'observation a été faite.

- **Réutilisation de vocabulaires RDF ;**

- ✓ La réutilisation des données aide grandement l'interopérabilité de vos données.
- ✓ La réutilisation ajoute de la crédibilité à votre schéma. Il montre qu'il a été publié avec soin et professionnalisme, ce qui favorise encore une fois sa réutilisation.
- ✓ La réutilisation est plus facile et moins chère. La réutilisation des classes et des propriétés de vocabulaires bien définis et correctement hébergés évite que vous ayez à reproduire cet effort.

- **Modéliser votre propre vocabulaire en tant que Schéma RDF :**

S'il n'ya pas de vocabulaire officiel de référence réutilisable et approprié pour décrire vos données, utilisez les conventions pour décrire votre propre vocabulaire:

- ✓ **Schéma RDF (RDFS).**
- ✓ **Langage d'ontologie Web (OWL).**

3.3.Schéma RDF (Ressource Description Framework Schéma) :

Le langage RDFS ajoute à RDF la possibilité de définir des hiérarchies de classes et de propriétés dont l'applicabilité et le domaine de valeurs peuvent être spécifiés à l'aide des attributs `rdfs:domain` et `rdfs:range`. Chaque domaine applicatif peut être ainsi associé à un schéma identifié par un préfixe particulier et correspondant à une URI. Les ressources instances sont ensuite décrites en utilisant le vocabulaire donné par les classes définies dans ce schéma. Les applications peuvent alors leur donner une interprétation opérationnelle. Cependant, on peut noter que RDFS n'intègre pas en tant que tel des capacités de raisonnement mais plutôt des solutions de base de données dédiées à RDFS apparaissant comme l'architecture Sésame à laquelle est associé le langage de requête SPARQL [1].

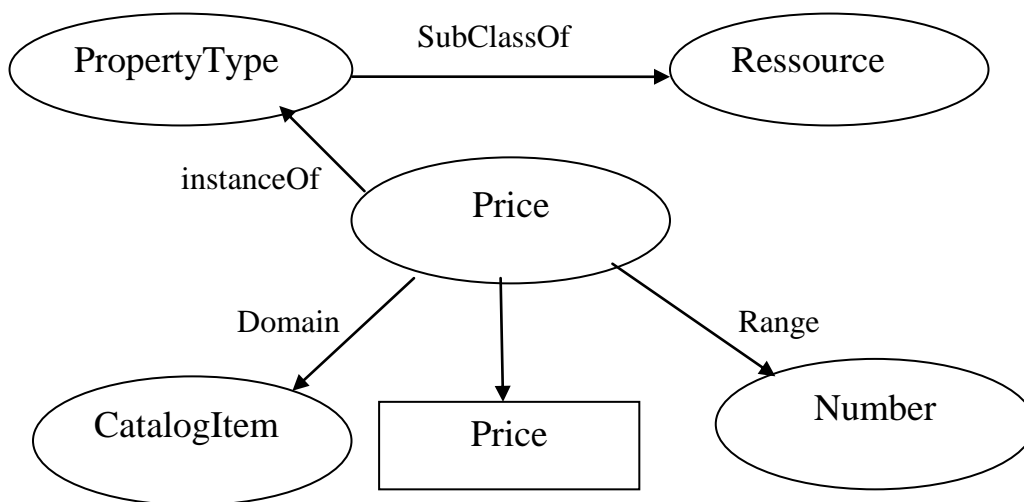


Figure 7 : Exemple schéma RDF.

4. Langages OWL:

Web Ontology Language est un langage de représentation des connaissances construit sur le modèle de données de RDF. Il fournit les moyens pour définir des ontologies web structurées. Sa deuxième version est devenue une recommandation du W3C fin 2012 [20].

Le langage OWL est basé sur les recherches effectuées dans le domaine de la logique de description. Il peut être vu en quelque sorte comme un standard informatique qui met en œuvre certaines logiques de description, et permet à des outils qui comprennent OWL de travailler avec ces données, de vérifier que les données sont cohérentes, de déduire des connaissances nouvelles ou d'extraire certaines informations de cette base de données. Il permet notamment de décrire des ontologies, c'est-à-dire qu'il permet de définir des terminologies pour décrire des domaines concrets. Une terminologie se constitue de concepts et de propriétés (aussi appelés « rôles » en logiques de description). Un domaine se compose d'instance de concepts.

Le langage OWL fournit des mécanismes pour créer tous les composants d'une ontologie : classes, instances, propriétés et axiomes. OWL repose également sur la syntaxe des triplets RDF et réutilise certaines des constructions RDFS. Comme en RDFS, les classes peuvent avoir des sous-classes, fournissant ainsi un mécanisme pour le raisonnement et l'héritage des propriétés [21]. Par contre, en OWL, on distingue :

1) **les propriétés objet (object property)**, i.e. Les relations, qui relient des instances de classes à d'autres instances de classes. C'est l'équivalent des triplets RDF dont l'objet est une ressource.

2) **les propriétés type de données (datatype property)**, i.e. Les attributs, qui relient des instances de classes à des valeurs de types de données (nombres, chaînes de caractères,...). C'est l'équivalent des triplets RDF dont l'objet est une valeur littérale. Les axiomes fournissent de l'information au sujet des classes et des propriétés, spécifiant par exemple l'équivalence entre deux classes.

4.1. Une extension de RDFS :

Malheureusement, bien que l'utilisation d'extensions de RDFS soit plus pertinente en théorie pour décrire le monde, il subsiste des problèmes pour les utiliser dans un contexte opérationnel. Par exemple, deux bases de données peuvent ne pas avoir le même comportement avec la même

ontologie même si l'implémentation des bases de données sont correcte Ces défauts sont inacceptables dans un système en production. Dès lors, les éditeurs de bases de données n'implémentent pas ou peu ces extensions et les inférences induites sont difficilement prévisibles sans avoir testé au préalable ces bases de données

Les développeurs d'ontologies doivent s'intéresser à la compatibilité des extensions de RDFS sur les bases de données cibles durant l'implémentation de leur ontologie sans quoi ce travail de description risque d'être inutile et alourdira inutilement l'accès aux données via SPARQL à travers le système d'information qui hébergera cette ontologie.

4.2. Les trois niveaux d'owl :

Le langage OWL est assez complexe, voila pourquoi il se compose de trois sous langages qui proposent une expressivité croissante, chacun conçu pour des communautés de développeurs et des utilisateurs spécifiques : OWL Lite, OWL DL, OWL Full. Chacun est une extension par rapport à son prédécesseur plus simple:

- a. **OWL Lite :** est le sous langage de OWL le plus simple. Il répond à des besoins de hiérarchie de classification et de fonctionnalités de contrainte simples de cardinalité 0 ou 1. Une cardinalité 0 ou 1 correspond à des relations fonctionnelles, par exemple, une personne a une adresse. Toutefois, cette personne peut avoir un ou plusieurs prénoms, OWL Lite ne suffit donc pas pour cette situation.
- b. **OWL DL :** est plus complexe que OWL Lite, permettant une expressivité bien plus importante. OWL DL est fondé sur la logique de descriptions (d'où son nom, OWL Description Logics) et conférant donc à OWL DL son adaptation au raisonnement automatisé. Malgré sa complexité relative face à OWL Lite, OWL-DL garantit la complétude des raisonnements (toutes les inférences sont calculables) et leur décidabilité (leur calcul se fait en une durée finie).
- c. **OWL Full :** est la version la plus complexe d'OWL, mais également celle qui permet le plus haut niveau d'expressivité. Le langage OWL Full se destine aux personnes souhaitant une expressivité maximale, ainsi que la liberté syntaxique de RDF, mais sans garantir la complétude et la décidabilité des calculs liés à l'ontologie. En OWL Full, une classe peut également être un individu, il n'y a pas de séparation des types. C'est pour cela que la calculabilité ne peut être garantie. Il est à noter que entre ces trois sous langage il existe une dépendance de nature hiérarchique. En effet, toute ontologie OWL Lite valide est également une ontologie OWL DL valide, et toute ontologie OWL DL valide est également une ontologie OWL Full valide [21] .

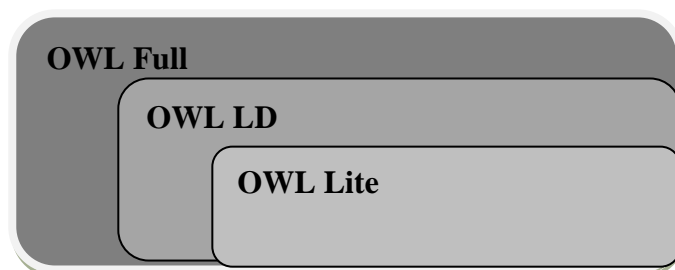


Figure 8 : Représenté de trois niveaux OWL (OWL1 Lite \subset OWL1 DL \subset OWL1 Full).

5. Ontologies:

La notion d'ontologie a été abordée pour la première fois par John McCarthy dans le domaine de l'intelligence artificielle (IA). Il affirmait déjà en 1980 que les concepteurs des systèmes intelligents fondés sur la logique devraient d'abord énumérer tout ce qui existe [40]. Cette approche présentée par John McCarthy n'est pas la seule puisque par la suite plusieurs définitions ont été proposées par d'autres auteurs du domaine :

Définition 1 : En intelligence artificielle, la définition communément admise d'une ontologie a été donnée par T. Gruber où il décrit une ontologie comme une spécification explicite d'une conceptualisation. T. Gruber défend ce point de vue comme suit : " An ontology is a specification of a conceptualisation. That is, an ontology is a description (like a formal specification of a program) of the concepts and Relationship That Can exist for an agent or a community of agent " [41].

Définition 2 : Après, afin de compléter le sens philosophique originel, N. Guarino a introduit la notion d'ontologie formelle, qui est définie en tant que modélisation conceptuelle, ou une représentation de cette modélisation. " Une ontologie est un accord sur une conceptualisation partagée et éventuellement partielle " [42].

Définition 3 : De même, M. Uschold définit une ontologie comme une description formelle d'entités et de leurs propriétés, relations, contraintes et comportements. De plus, les auteurs ont introduit, dans [43], la notion de l'ontologie explicite "An explicit ontology may take a variety of forms, but necessarily it will include a vocabulary of terms and some specification of their meaning".

Définition 4 : C. Roche a donné une définition générique et simple "Une ontologie est une conceptualisation d'un domaine à laquelle sont associés un ou plusieurs vocabulaires de termes. Les concepts se structurent en un système et participent à la signification des termes. Une ontologie est définie pour un objectif donné et exprime un point de vue partagé par un groupe de personne. Une ontologie s'exprime dans un langage (représentation) qui repose sur une théorie (sémantique) qui garantit des propriétés de l'ontologie en termes de consensus, cohérence, réutilisation et partage" [44].

Définition 5 : Pour Oberle, les ontologies, en tant que systèmes de représentation, peuvent être considérées au même titre que des systèmes de modélisation conceptuelle, tels que le modèle entité-relation de Chen (1976) ou le modèle UML de Rumbaugh, Jacobson et Booch (1999). Cependant, l'ontologie se distingue de ces systèmes de représentation par les points suivants [45][46]:

- l'idée primaire des ontologies est d'établir un certain niveau de consensus autour d'un vocabulaire afin de faciliter l'échange d'informations entre applications ;
- les ontologies sont formalisées au moyen d'un système de représentation logique dont la sémantique est spécifiée de façon non ambiguë ;
- l'ontologie est exprimée au moyen d'un langage de représentation qui, à l'état

d'exécution, sert de représentation à des applications de recherche et de raisonnement. Dans **la figure 9**, nous présentons une synthèse des principales définitions d'une ontologie, du point de vue informatique, schématisée dans le langage graphique de Modélisation par Objets Typés (MOT), dont on trouve une description détaillée dans [46][47].

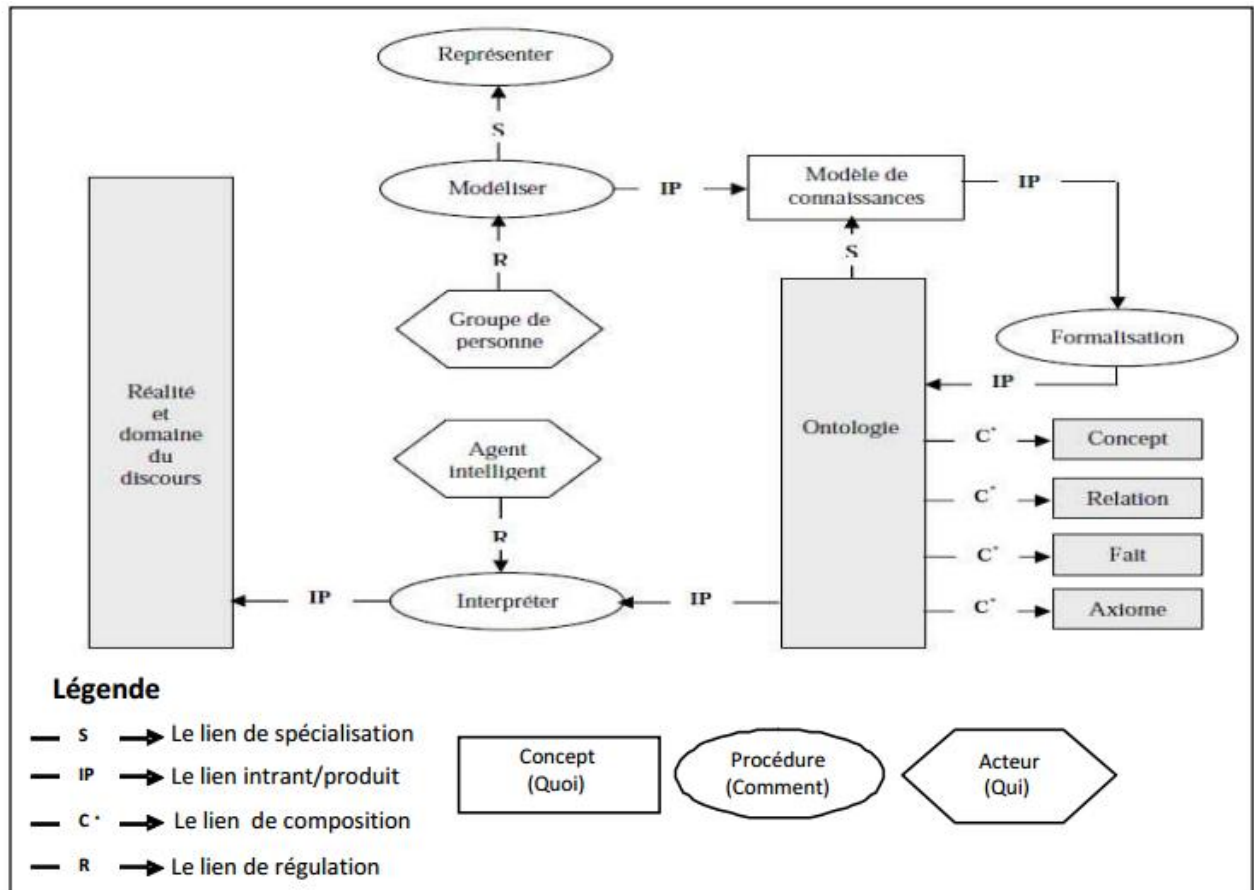


Figure 9 : Définition d'une ontologie représentée.

5.1. Objectif :

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné, qui peut être réel ou imaginaire.

Les ontologies sont employées dans l'intelligence artificielle, le Web sémantique, le génie logiciel, l'informatique biomédicale ou encore l'architecture de l'information comme une forme de représentation de la connaissance au sujet d'un monde ou d'une certaine partie de ce monde. Les ontologies décrivent généralement :

Individus	les objets de base
Classes	ensembles, collections, ou types d'objets
attributs	propriétés, fonctionnalités, caractéristiques ou paramètres que les objets peuvent posséder et partager
Relations	les liens que les objets peuvent avoir entre eux
Evénements	changements subis par des attributs ou des relations

5.2. Principe :

Selon Gruber, « l'ontologie est une spécification explicite d'une conceptualisation », c'est-à-dire qui permet de spécifier dans un langage formel les concepts d'un domaine et leurs relations.

5.3. Evaluation d'ontologie :

D'après Gruber [18], cinq critères permettent de mettre en évidence des aspects importants d'une ontologie :

1. **La clarté :** la définition d'un concept doit faire passer le sens voulu du terme, de manière aussi objective que possible (indépendante du contexte). Une définition doit de plus être complète (c'est-à-dire définie par des conditions à la fois nécessaires et suffisantes) et documentée en langage naturel.
2. **La cohérence :** rien qui ne puisse être inféré de l'ontologie ne doit entrer en contradiction avec les définitions des concepts (y compris celles qui sont exprimées en langage naturel).
3. **L'extensibilité :** les extensions qui pourront être ajoutées à l'ontologie doivent être anticipées. Il doit être possible d'ajouter de nouveaux concepts sans avoir à toucher aux fondations de l'ontologie.
4. **Une déformation d'encodage minimale :** une déformation d'encodage a lieu lorsque la spécification influe sur la conceptualisation (un concept donné peut être plus simple à définir d'une certaine façon pour un langage d'ontologie donné, bien que cette définition ne corresponde pas exactement au sens initial). Ces déformations doivent être évitées autant que possible.
5. **Un engagement ontologique minimal :** le but d'une ontologie est de définir un vocabulaire pour décrire un domaine, si possible de manière complète ; ni plus, ni moins. Contrairement aux bases de connaissances par exemple, on n'attend pas d'une ontologie qu'elle soit en mesure de fournir systématiquement une réponse à une question arbitraire sur le domaine. Toujours selon Gruber, « l'engagement ontologique peut être minimisé en spécifiant la théorie la plus faible (celle permettant le plus de modèles) couvrant un domaine ; elle ne définit que les termes nécessaires pour partager les connaissances consistantes avec cette théorie ».

5.4. Types d'ontologie :

On identifie trois types d'ontologies selon un niveau décroissant d'abstraction :

- a. **Les ontologies globales (*Top-Level Ontology*) :** présentent un plus haut niveau d'abstraction et de généralité, sont les ontologies formelles, car elles sont issues d'un développement systématique, rigoureux et axiomatique de la logique de toutes les formes et modes d'existence. L'adoption de principes rigoureux dans la conception de l'ontologie formelle répond au besoin de disposer de connaissances pouvant être partagées et transférées d'un contexte à l'autre. Elles sont dédiées à des utilisations générales (ex : WordNet). Une ontologie formelle est donc une théorie des distinctions formelles entre les éléments d'un domaine, indépendamment de leur réalité (Guarino, 1997).
- b. **Les ontologies de domaine, ou dédiées à une tâche plus spécifique :** limitées à la représentation de concepts dans des domaines donnés (géographie, médecine, écologie, etc.) et qui spécialise les concepts de l'ontologie globale.
- c. **Les ontologies d'application (Guarino, 1998) :** offrent le plus fin niveau de spécificité, c'est-à-dire qu'elles sont dédiées à un champ d'application précis à l'intérieur d'un domaine et décrivent le rôle particulier des entités de l'ontologie de domaine dans ce champ. Par exemple, l'ensemble des spécifications sur la forêt de Montmorency constitue une ontologie d'application qui spécifie les concepts généraux pouvant provenir d'une ontologie de domaine forestier générale.

5.5. Langage d'ontologie :

Le langage de spécification est l'élément central sur lequel repose l'ontologie.

La plupart de ces langages se basent sur ou sont proches de la logique du premier ordre, et représentent donc les connaissances sous forme d'assertion (sujet, prédicat, objet). Ces langages sont typiquement conçus pour s'abstraire des structures de données et se concentrer sur la sémantique [19]. Parmi les formalismes les plus employés se basant sur la logique des prédicats, on retrouve des langages comme N3 ou N-Triple. On peut aussi évoquer le langage DEF-*

Par ailleurs, dans le cadre de ses travaux sur le Web sémantique, le W3C a mis en place en 2002 un groupe de travail consacré au développement de langages standards pour modéliser des ontologies utilisables et échangeables sur le Web. S'inspirant de langages précédents comme DAML+OIL et des fondements théoriques des logiques de description, ce groupe a publié en 2004 une recommandation définissant le langage OWL (*Web Ontology Language*), fondé sur le standard RDF et en spécifiant une syntaxe XML. Plus expressif que son prédécesseur RDFS, OWL a rapidement pris une place prépondérante dans le paysage des ontologies et est désormais, de facto, le standard le plus utilisé.

Bien que développé pour la représentation des vocabulaires contrôlés et structurés (thésaurus), SKOS peut être utilisé pour élaborer et gérer des ontologies légères multilingues.

6. SPARQL (Protocol and RDF Query Language):

SPARQL est un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles à travers Internet. Son nom est un acronyme récursif qui signifie SPARQL Protocol and RDF Query Language.

SPARQL est l'équivalent de SQL car comme en SQL, on accède aux données d'une base de données via ce langage de requête alors qu'avec SPARQL, on accède aux données du Web des données. Cela signifie qu'en théorie, on pourrait accéder à toutes les données du Web avec ce standard. L'ambition du W3C est d'offrir une interopérabilité non pas seulement aux niveaux des services, comme avec les services Web, mais aussi aux niveaux des données structurées ou non qui sont disponibles à travers l'Internet.

Ce standard a été créé par le groupe de travail DAWG (RDF Data Access Working Group) du W3C (Consortium World Wide Web). SPARQL est considéré comme l'une des technologies clés du Web sémantique et le 15 janvier 2008, la version 1.0 est devenue une recommandation officielle du W3C. La version 1.1 permettra d'enregistrer des données et de fusionner des données de sources différentes. La version 1.1 est devenue depuis le 21 mars 2013 une recommandation.

Les implémentations de SPARQL au sein de triple stores se multiplient. « SPARQL fera une énorme différence », selon Tim Berners-Lee dès mai 2006 [22].

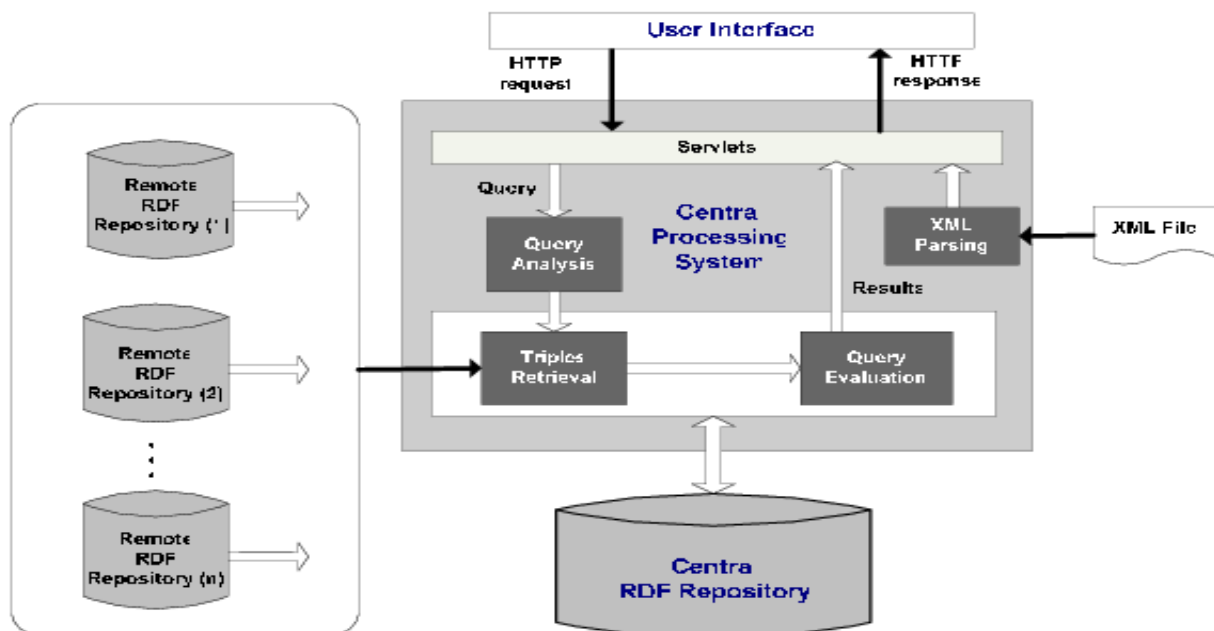


Figure 10 : Architecture du mécanisme de requête SPARQL pour les référentiels RDF distribués [23].

6.1. Caractéristique :

SPARQL est adapté à la structure spécifique des graphes RDF, et s'appuie sur les triplets qui les constituent. En cela, il est différent du classique SQL (langage de requête qui est adapté aux bases de données de type relationnelles), mais s'en inspire clairement dans sa syntaxe et ses fonctionnalités. Il a aussi quelques traits de ressemblance mineurs avec Prolog [22].

SPARQL permet d'exprimer des requêtes interrogatives ou constructives :

- une requête **SELECT**, de type interrogative, permet d'extraire du graphe RDF un sous-graphe correspondant à un ensemble de ressources vérifiant les conditions définies dans une clause **WHERE**.
- une requête **CONSTRUCT**, de type constructive, engendre un nouveau graphe qui complète le graphe interrogé.

Par exemple sur un graphe RDF contenant des informations généalogiques, on pourra par une requête SELECT trouver les parents ou grands-parents d'une personne donnée, et par des requêtes CONSTRUCT ajouter des relations frère-sœur, cousin-cousine, oncle-neveu, qui ne seraient pas explicitement déclarées dans le graphe initial.

6.2. Structure de requête SPARQL :

La figure 11 représente la structure d'une requête SPARQL :

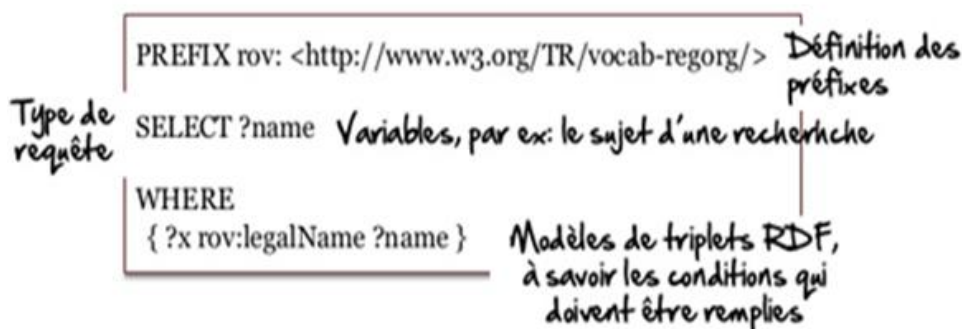


Figure 11 : Structure de la requête SPARQL.

6.3. Processus de requête SPARQL :

La figure 12 représente le processus de requête SPARQL :

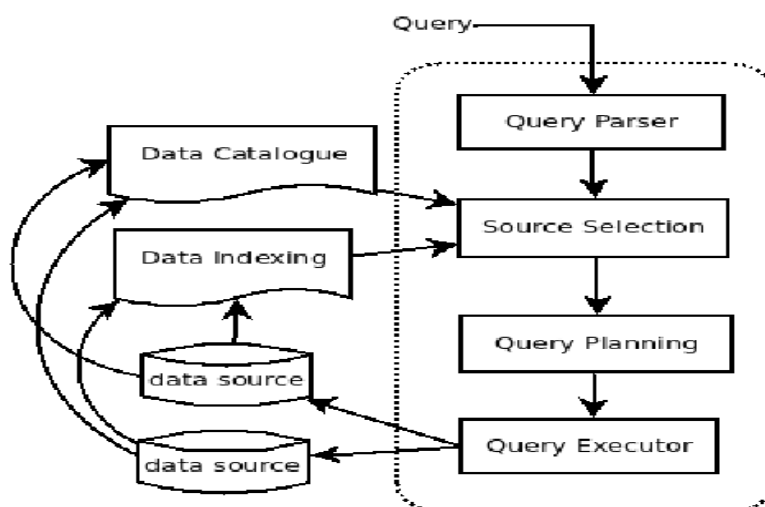


Figure 12 : le processus de requête SPARQL.

6.4. Mise à jour SPARQL :

Peut être utilisé pour :

- L'ajout de données (**INSERT**).
- La suppression de données (**DELETE**).
- Le chargement de graphe RDF (**LOAD / LOAD ... INTO**).
- L'effacement d'un graphe RDF (**CLEAR GRAPH**).
- Créer des graphes RDF (**CREATE GRAPH**).
- Retirer des graphes RDF (**DROP GRAPH**).
- Copier des graphes RDF (**COPY GRAPH ... TO GRAPH**).
- Déplacer des graphes RDF (**MOVE GRAPH ... TO GRAPH**).
- Ajouter des graphes RDF (**ADD GRAPH TO GRAPH**).

Conclusion:

Ce chapitre introductif nous a permis de décrire quelques généralités autour de notre projet. Et ce en donnant un aperçu sur les technologies du Web de données et conclus que :

- ✓ Les données liées sont un ensemble de principes de conception pour le partage de données lisibles par machine.
- ✓ Les données liées et les données ouvertes ne sont pas la même chose.
- ✓ URI, RDF, OWL et SPARQL forment la couche de fondation pour les données liées.

Dans ce qui suit nous décrivons l'analyse et la conception des différentes fonctionnalités du projet que nous allons développer.

Chapitre 2 : Analyse et conception d'un système d'apprentissage basé sur linked data

L'analyse et la conception sont les phases clés dans le cycle de développement d'un logiciel et de la réussite d'un projet. Ce chapitre se compose de deux phases, la première étape est la conception et la production du système, et la deuxième partie consiste à réaliser une ontologie de web sémantique et à transformer la requête SPARQL en une question dans le langage naturel. L'objectif est de trouver et d'installer une nouvelle technologie pour faciliter l'apprentissage de l'apprenant. Ainsi dans ce chapitre nous présentons les étapes d'analyse et de conception qui ont menés à la réalisation de ce projet.

1. Etape1 : Analyse des besoins

L'analyse du besoin n'est pas complexe à comprendre, elle fait appel au bon sens, toutefois sa mise en œuvre repose sur un enchaînement qu'il faut respecter. Si on veut aboutir à un résultat satisfaisant et compréhensible par tous. Dans cette section, nous allons identifier un ensemble de besoins fonctionnels indispensables pour tous les acteurs, mais aussi des besoins non fonctionnels.

1.1. Besoins fonctionnelles :

Les besoins fonctionnels se rapportent aux fonctionnalités que doit fournir l'application pour satisfaire les attentes des utilisateurs.

1.2. Besoins non fonctionnelles :

Les besoins non fonctionnels sont indispensables et permettent l'amélioration de la qualité de l'application. En effet cette dernière doit aussi répondre aux exigences ci dessous :

- Le système doit être convivial et intuitif, facile dans la navigation.
- L'intégrité du système : les informations ne peuvent être modifiées que par les personnes qui sont autorisés.
- La sécurité : le système doit disposer d'un minimum de sécurité.
- Le système doit être robuste et ne doit en aucun cas alourdir son utilisation, celle-ci doit être aisée.
- Le système doit être extensible pour mettre des améliorations futures.
- L'application doit pouvoir s'adapter aux différent terminaux (mobile, desktop, ...).

2. Spécifications fonctionnelles :

2.1. Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il modélise un service rendu par le système, sans imposer le mode de réalisation de ce service. Ils permettent de décrire l'interaction entre l'acteur et le système. L'idée forte est de dire que l'utilisateur d'un système logiciel a un objectif quand il utilise le système. Composé par : des acteurs sont des entités externes qui interagissent avec le système, comme une personne humaine ou un robot. Une même personne(ou robot) peut être plusieurs acteurs pour un système. On reconnaît 4 catégories :

- Les acteurs principaux (ex: Visiteur, Apprenant, Enseignant, ...etc.),
- Les acteurs secondaires (ex: Administrateur, ...etc.),
- Le matériel externe (ex : imprimante, ...etc.),
- Les autres systèmes (ex: serveur, ...etc.).

❖ Les acteurs de notre système sont :

✓ Administrateur :

C'est l'administrateur du système, il a le droit de faire toutes les actions affectées aux autres acteurs, la gestion des pages, plus la gestion des comptes des utilisateurs.

✓ Enseignant :

C'est l'acteur qui a le droit de faire toutes les actions affectées aux Apprenant ou enseignants (Annotations, documents pédagogiques, annonces, affichage,..., etc.), plus la gestion des espaces enseignant, la vérification et la validation en ligne.

✓ Apprenant :

Cet acteur peut consulter, connecté, discuter, et suivre le déroulement d'avancement (cours, annoter, faire des tests on ligne..., etc.)

✓ **Visiteur :**

Il a le droit de consulter les différentes pages qui sont existées (visite panoramique) en tant que visiteur n'a pas possibilité d'accéder à certaines fonctionnalités (cours, ..., etc.).

❖ **Description détaillée des diagrammes de cas d'utilisation :**

a. Gestion des administrateurs :

La figure 13 présente le diagramme de cas d'utilisation des administrateurs, tel que :

- Gérer les Documents pédagogiques :
- Gérer les annonces (ajouter, modifier, supprimer).
- Gérer les utilisateurs (Ajouter, Supprimer).
- Contrôler les annotations des apprenants de son groupe.

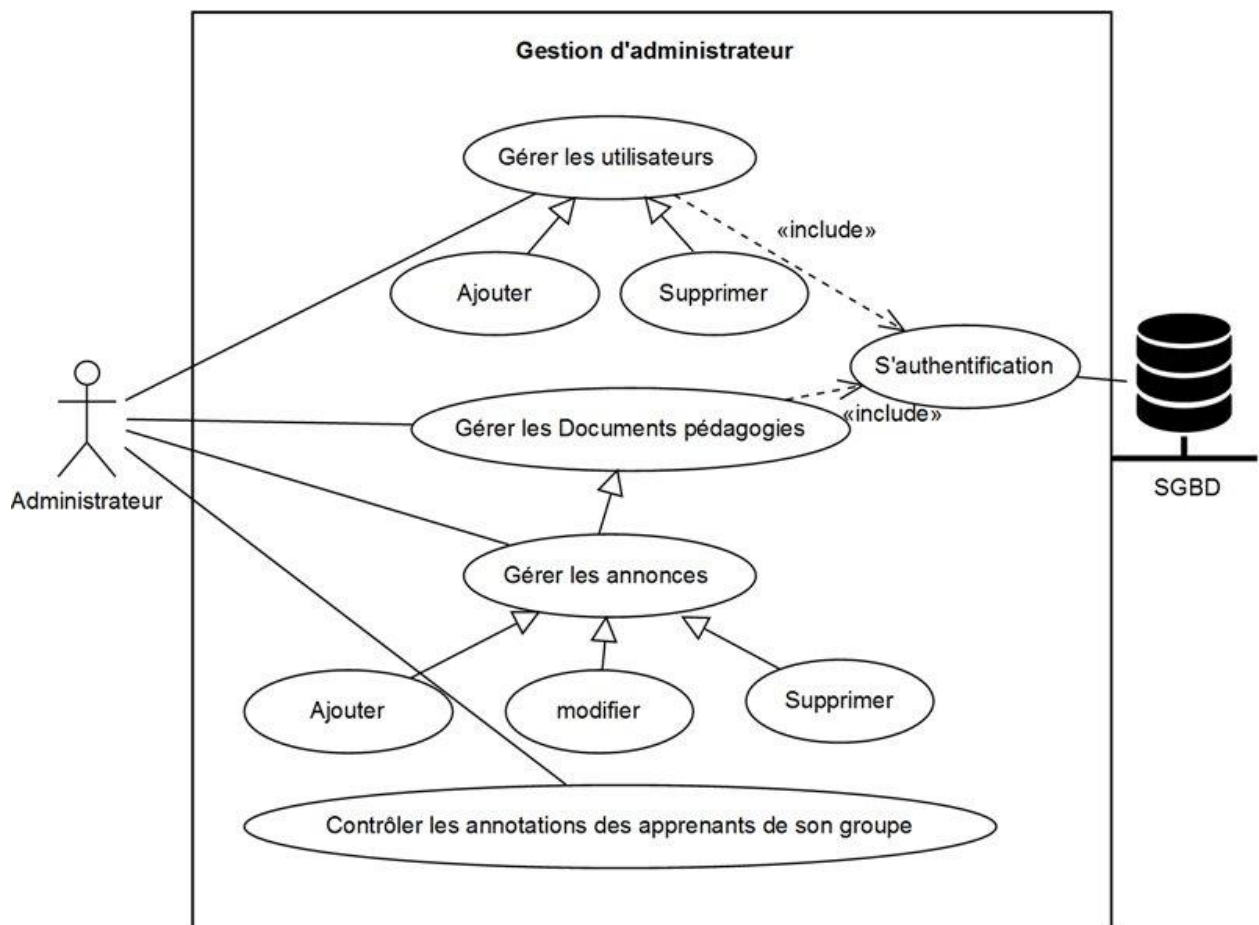


Figure 13 : diagramme de cas d'utilisation (gestion des administrateurs).

b. Gestion des Enseignants :

La figure 14 présente le diagramme de cas d'utilisation des enseignants, Tel que:

- Naviguer sur les pages permises.
- Déposer les cours (documents pédagogie : ajouter, modifier, supprimer).
- QCM en ligne (ajouter, modifier, supprimer).
- Ajouter groupe des apprenants.
- Publier des annonces de cours.
- Afficher les annotations personnelles ou partagés par groupe.
- Assister a forums.
- Se Connecté.



Figure 14 : diagramme de cas d'utilisation (gestion des enseignants).

c. Gestion des apprenants :

La figure 15 présente le diagramme de cas d'utilisation des apprenants, Tel que :

- Naviguer sur les pages permises.
- Consulter les documents pédagogique (DP) :
- Annoter les DP.
- Afficher annotation partagé.
- QCM en ligne.
- Assister à forums.
- Se connecté.

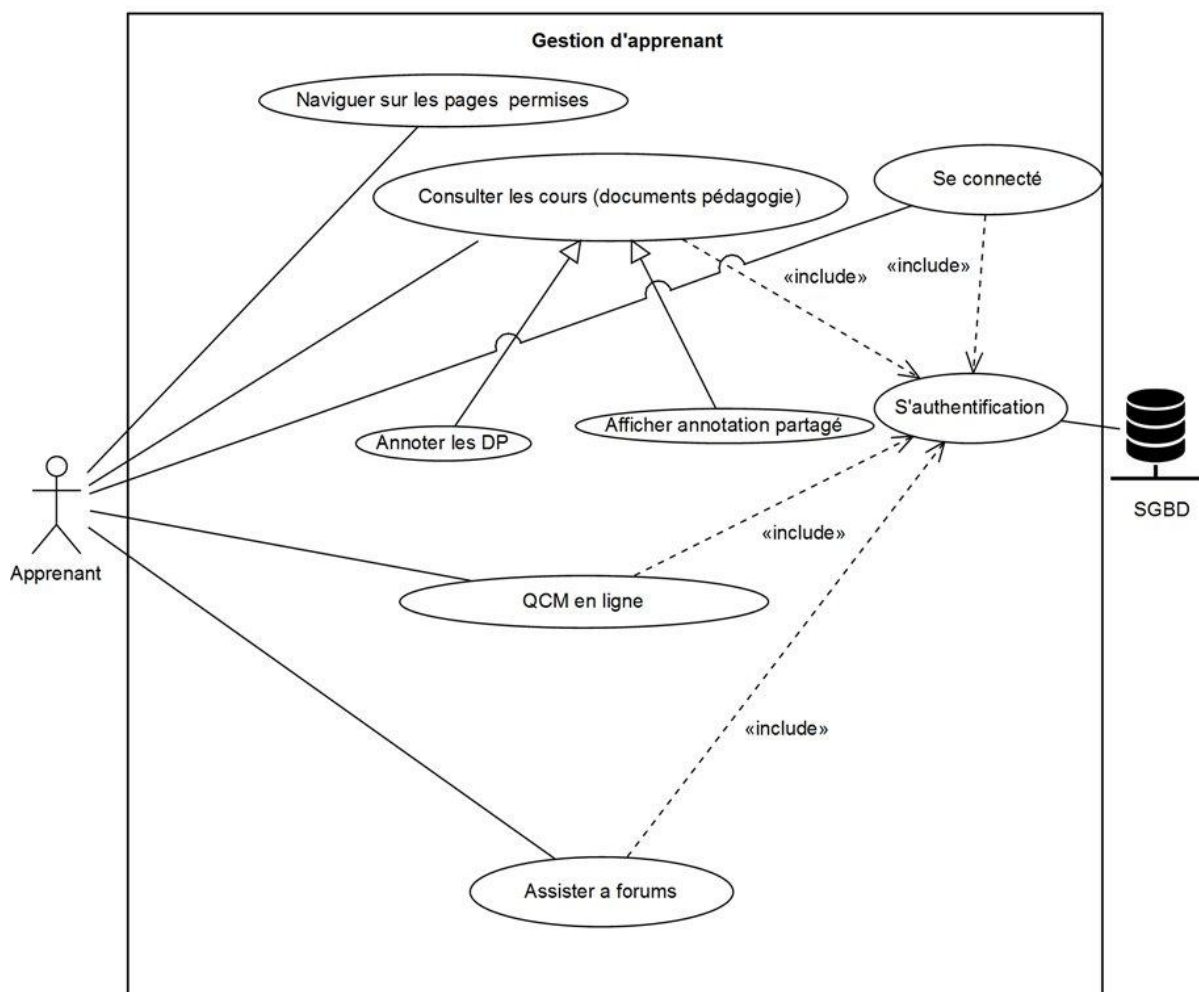


Figure 15 : diagramme de cas d'utilisation (gestion des apprenants).

d. Gestion visiteurs :

La figure 16 présente le diagramme de cas d'utilisation des visiteurs, Tel que :

- Consulter les pages.
- Consulter les annonces.
- Contactez l'administrateur.

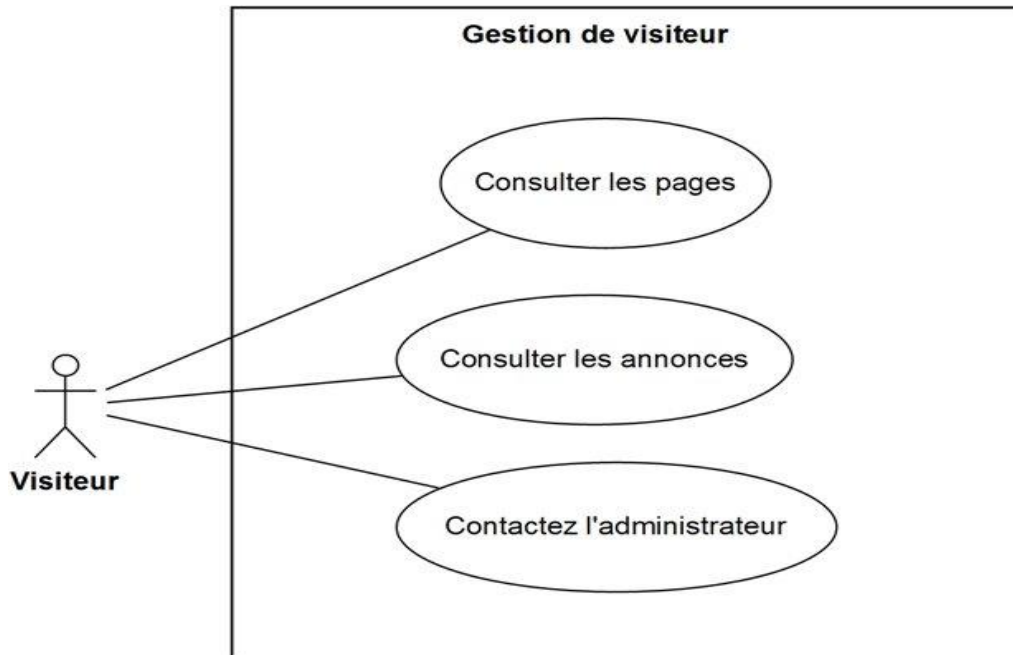


Figure 16 : diagramme de cas d'utilisation (gestion des visiteurs).

2.2. Diagramme d'activité :

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Il est recommandable pour exprimer une dimension temporelle sur une partie du modèle, à partir de classe sous de cas d'utilisation. Le diagramme d'activité est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis. Le diagramme d'activité est sémantiquement proche des diagrammes de communication (Appelés *diagramme de collaboration* en UML), ou d'état-transitions, ces derniers offrant une vision microscopique des objets du système.

❖ Description détaillée des diagrammes d'activité:

a. Diagramme d'activité des acteurs (Apprenants, enseignants, administrateurs) :

La figure 17 présente le diagramme d'activité des Acteurs (gestion de connexion, inscription et l'annotation), Tel que :

- L'enseignant et l'apprenant accèdent aux E-Learning.
- Ils demandent la connexion.
- S'ils sont inscrit ils vont entrer ses informations (Nom d'utilisateur et mot de passe) et commencer la connexion sur le site.
- Sinon, ils vont demandent une inscription.

- L'apprenant télécharger document pédagogique (vidéo).
- L'apprenant testé en ligne par un questionnaire.
- L'apprenant consulter DP et annoter par des questions.
- L'enseignant annoté (répond aux questions de l'apprenant), contrôle les annotations partagés et contrôle les annotations personnelles.
- L'administrateur ajouter modules, inscrit des enseignants et demande l'inscription
- Le système répondre automatique a notre annotations.

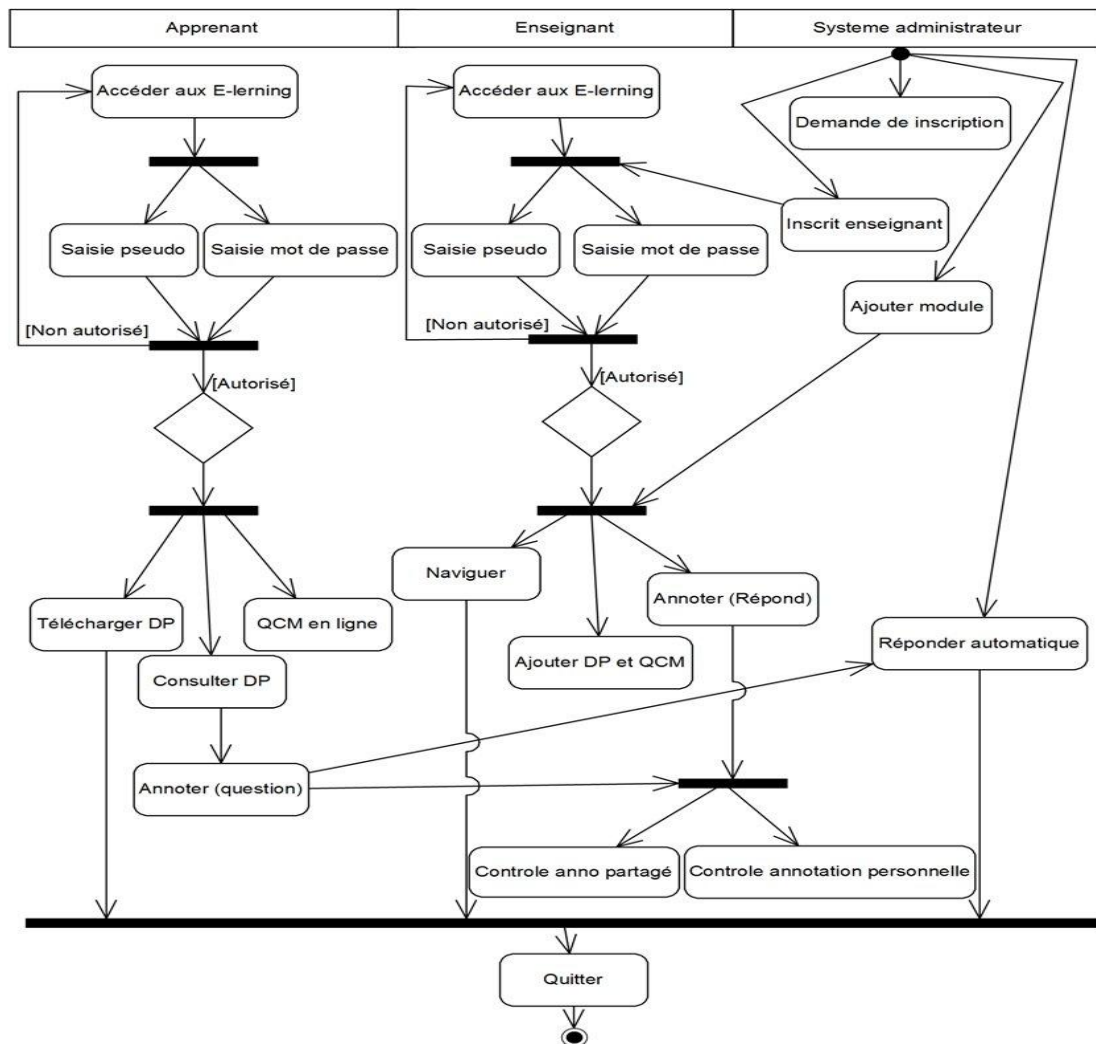


Figure 17 : diagramme d'activité (gestion des apprenants, des enseignants et des administrateurs (système)).

2.3. Conception détaillée :

Dans cette phase une nouvelle vue du modèle fait son apparition. Cette vue exprime Les Modules et les exécutable physiques sans aller à la réalisation concrète du système. Elle est basée sur :

- Les diagrammes de classe (Vue statique).
- Les diagrammes de séquence (Vue dynamique).

2.3.1. Diagramme de classe :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter Les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait la partie statique d'UML car il fait abstraction des aspects temporels et Dynamiques. La classe est un moule à objet : elle décrit les responsabilités, les comportements, les Caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces Caractéristiques lorsqu'ils sont instanciés. La classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble Par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en Plusieurs petits travaux simples. Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre En évidence des relations de parenté. D'autres relations sont possibles entre des classes, Chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

❖ Description détaillée des diagrammes de classes:

a. Gestion des administrateurs :

La figure 18 présente le diagramme de classe des administrateurs, Tel que :

Chaque administrateur est caractérisé par (Id, nom, prénom, username, email, login, password)

❖ Règle de gestion :

- Un administrateur a un est une seule connexion (login), et une connexion peut caractériser un ou plusieurs administrateurs.

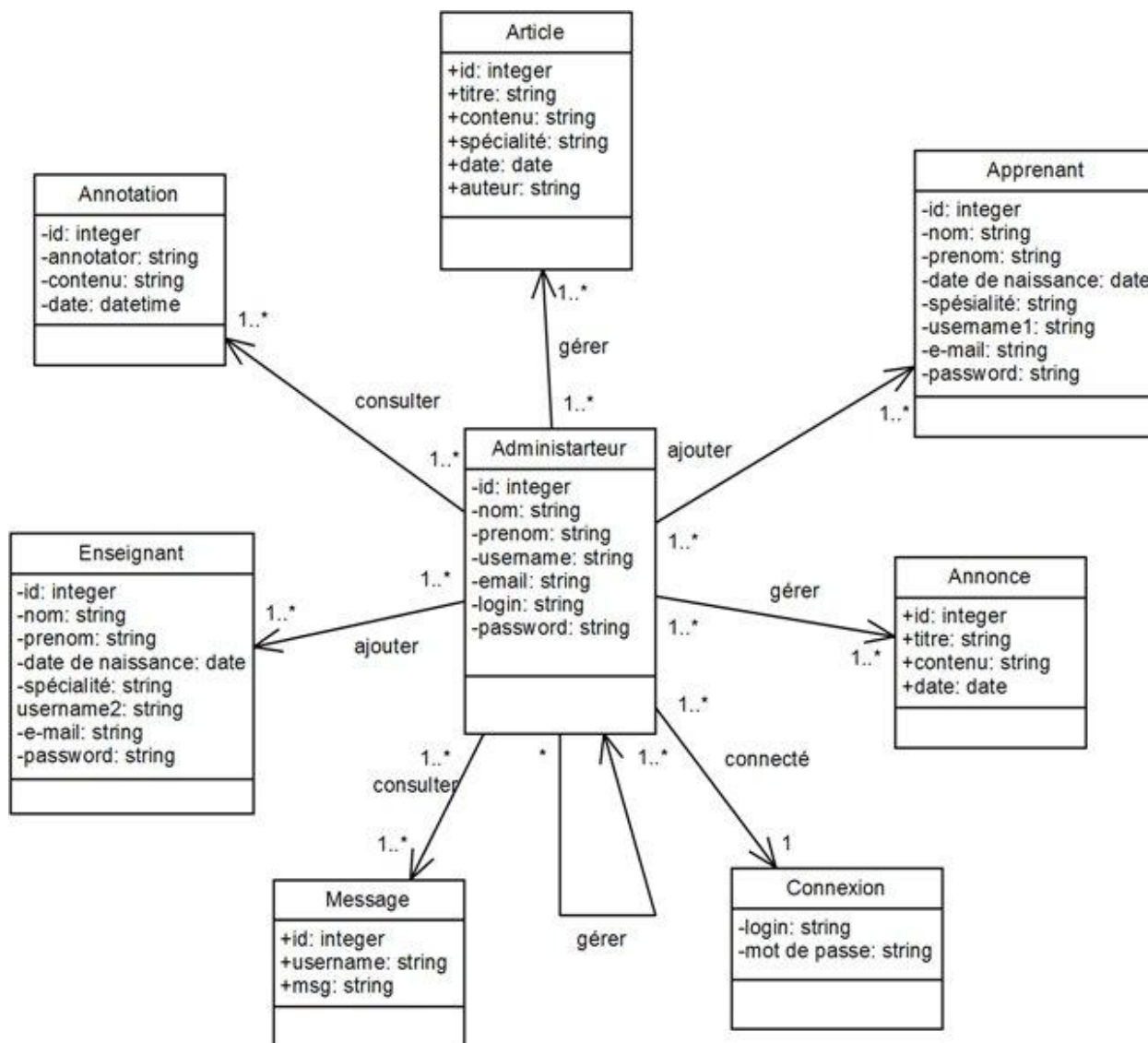


Figure 18 : Diagramme de classe (gestion d'administrateur).

b. Gestion des Enseignants :

La figure 19 présente le diagramme de classe des enseignants, Tel que :

Chaque enseignant est caractérisé par (Id, Nom, Prénom, Date de naissance, username2, Spécialité, E-mail, mot de passe)

❖ Règles de gestion :

- Un enseignant a un est une seule connexion (login), et une connexion peut caractériser un ou plusieurs enseignants.
- Enseignant a une seule spécialité, et une spécialité peut être concerné plusieurs enseignants.

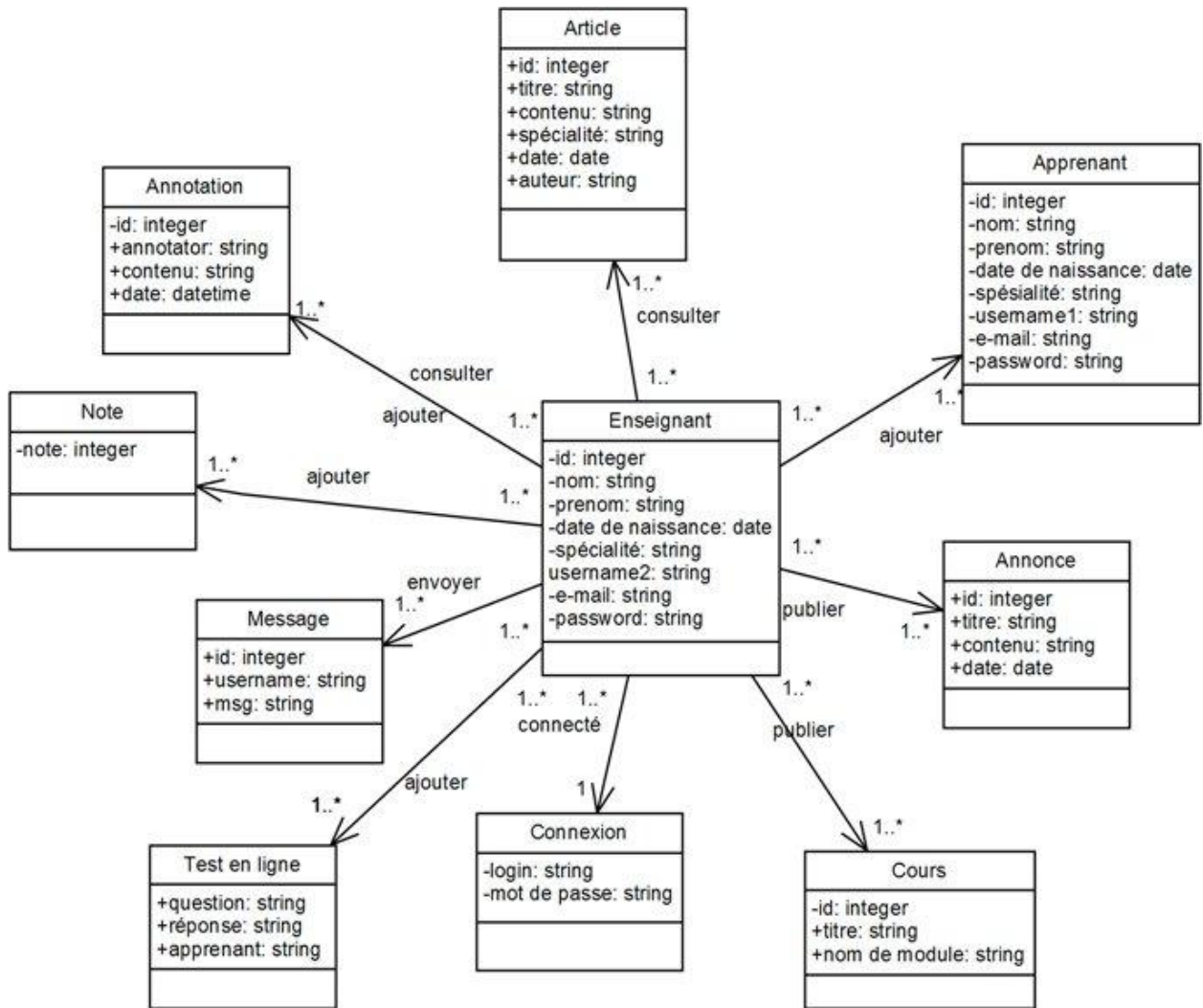


Figure 19 : Diagramme de classe (gestion d'enseignant).

c. Gestion des apprenants :

La figure 20 présente le diagramme de classe des apprenants, Tel que :

Chaque apprenant est caractérisé par (Id, Nom, Prénom, Date de naissance, Spécialité, E-mail, username1, mot de passe).

❖ Règle de gestion :

Un apprenant a un est une seule connexion (login), et une connexion peut caractériser un (ou) plusieurs apprenants.

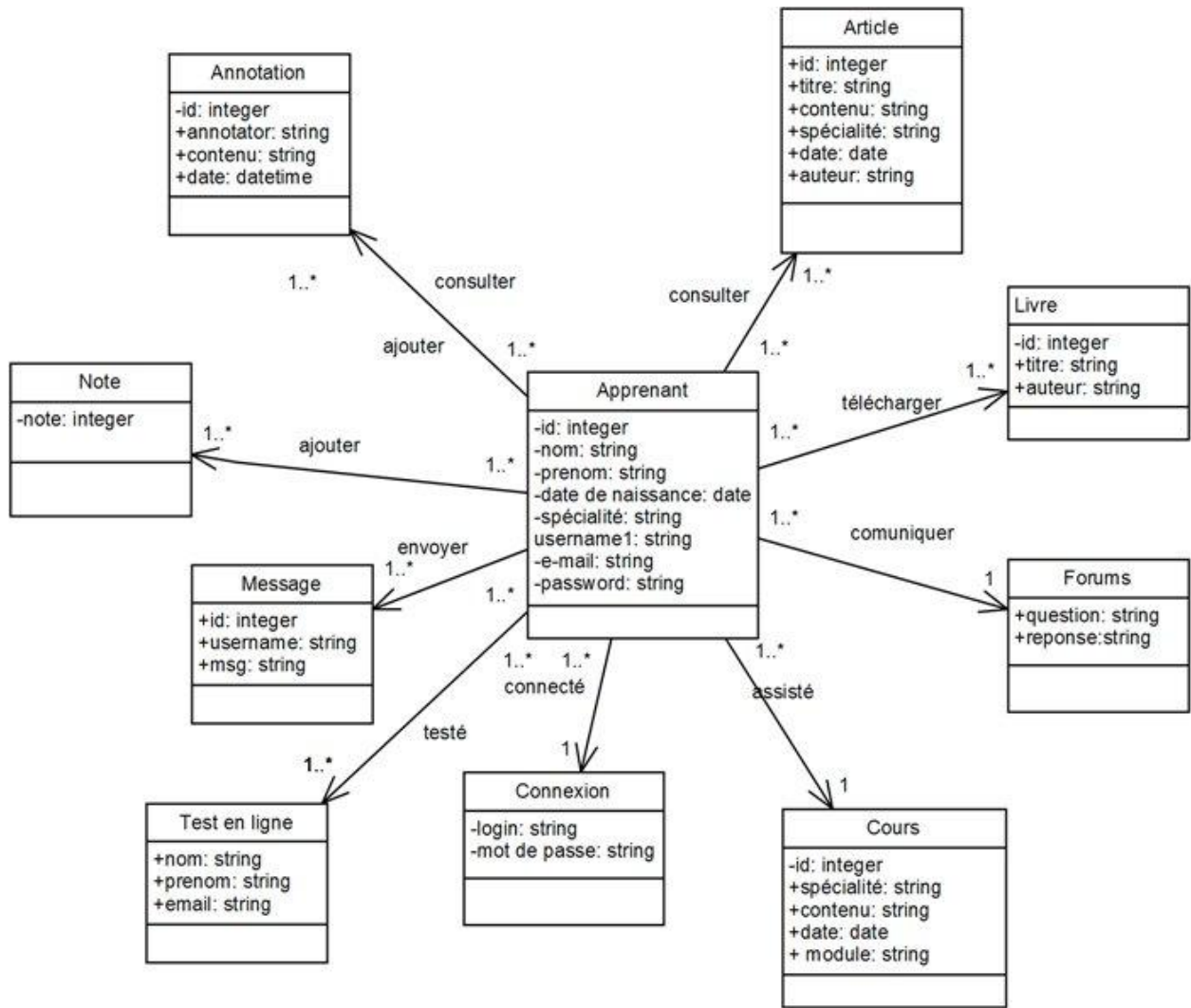


Figure 20 : Diagramme de classe (gestion d'apprenant).

d. Gestion des Visiteurs :

La figure 21 présente le diagramme de classe de visiteur, Tel que :

Chaque visiteur est caractérisé par (Id, Nom, Prénom).

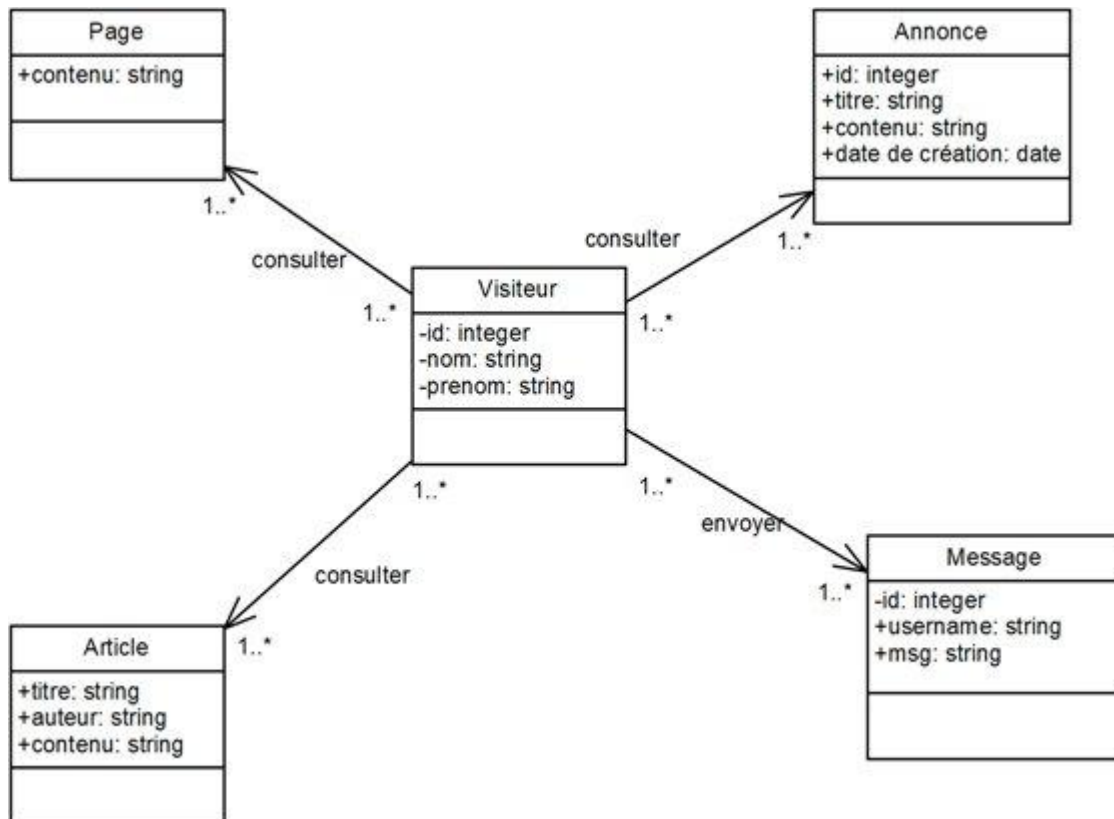


Figure 21 : Diagramme de classe (gestion de visiteur).

2.3.2. Diagramme de séquence :

Les diagrammes de séquence expriment les interactions qui surviennent dans une séquence de Temps. En particulier ils montrent la participation d'objets dans les interactions et les Messages qu'ils échangent dans un intervalle de temps. Ils ne montrent pas les associations Entre les objets. Les composants de ces diagrammes sont : ligne de vie, acteur, objets, message, action.

❖ Description détaillée des diagrammes de séquence:

a. Gestion d'apprenant :

La figure 22 présente le diagramme de séquence des apprenants, Tel que :

- L'apprenant accéder au page premise et demande la page connexion.
- L'apprenant demande la page connexion
- Serveur lui affichera boite de connexion.
- L'apprenant entrer ses informations.
- Serveur envoi donnée au système.
- Système vérifier les données.
- Système retourner a la page de connexion qu'il y a une erreur.
- Système vérifier les données et lui donnera l'autorisation de consulter le cours.
- L'apprenant consulter les documents pédagogies.

- Le serveur envoyé la demande.
- Il lui accepté.
- L'apprenant consulter la vidéo et annoter.
- Serveur envoi les annotations au système.
- Système affichera les réponses d'annotations.
- L'apprenant déconnecté.

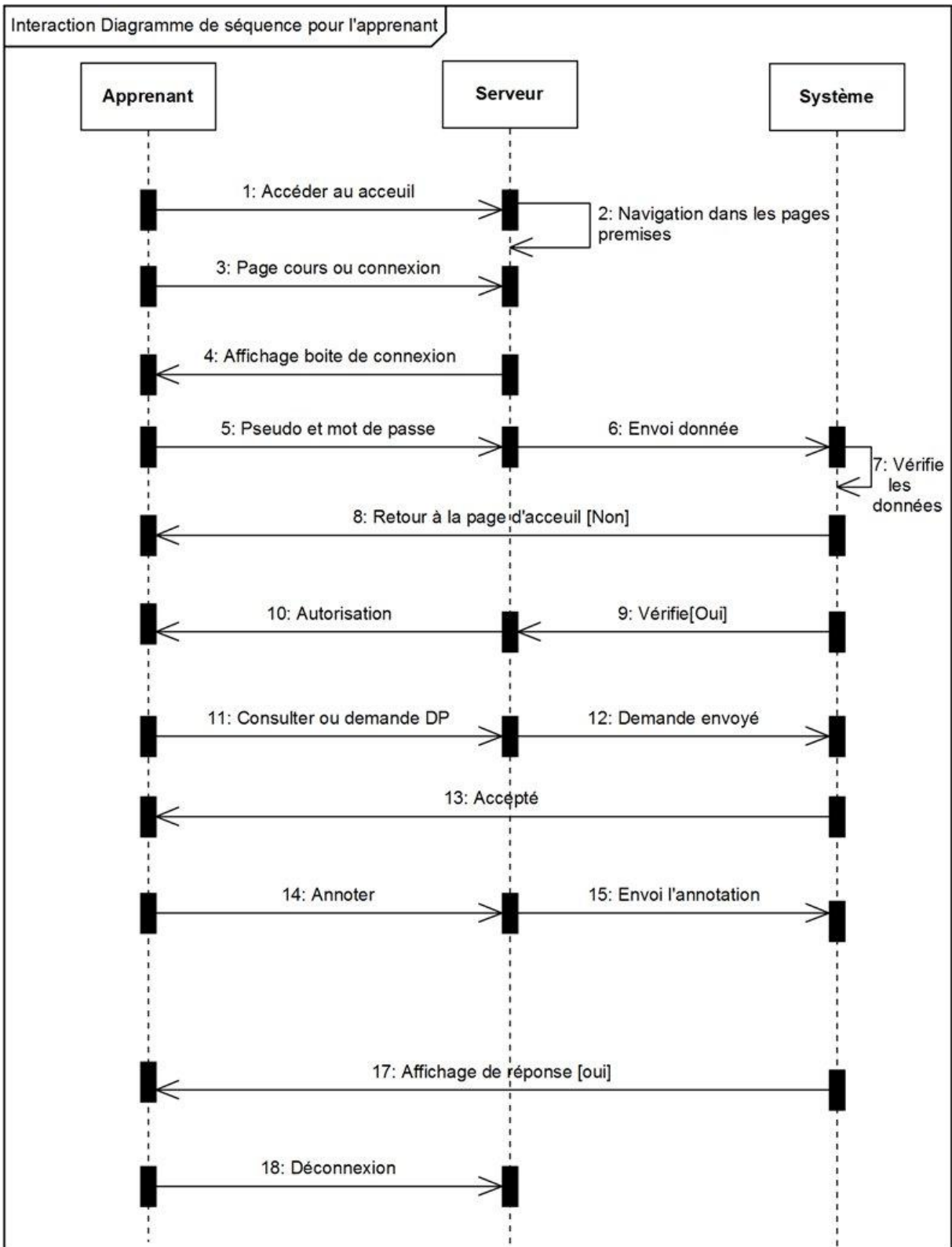


Figure 22 : diagramme de séquence (gestion des apprenants).

b. Gestion d'enseignant :

La figure 23 présente le diagramme de séquence des enseignants, Tel que :

- L'enseignant accéder au page premise et demande la page connexion.
- L'enseignant demande la page connexion
- Serveur lui affichera boite de connexion.
- L'enseignant entrer ses informations.
- Serveur envoi donnée au système.
- Système vérifier les données.
- Système retourner a la page de connexion qu'il y a une erreur.
- Système vérifier les données et lui donnera l'autorisation de consulter le cours.
- L'enseignant mettre a jour les documents pédagogiques.
- Le serveur enregistré les modifications.
- Système affichera les annotations des apprenants.
- L'enseignant annoté.
- Serveur enregistré les annotations.
- Système affichera les annotations personnelles.
- L'enseignant déconnecté.

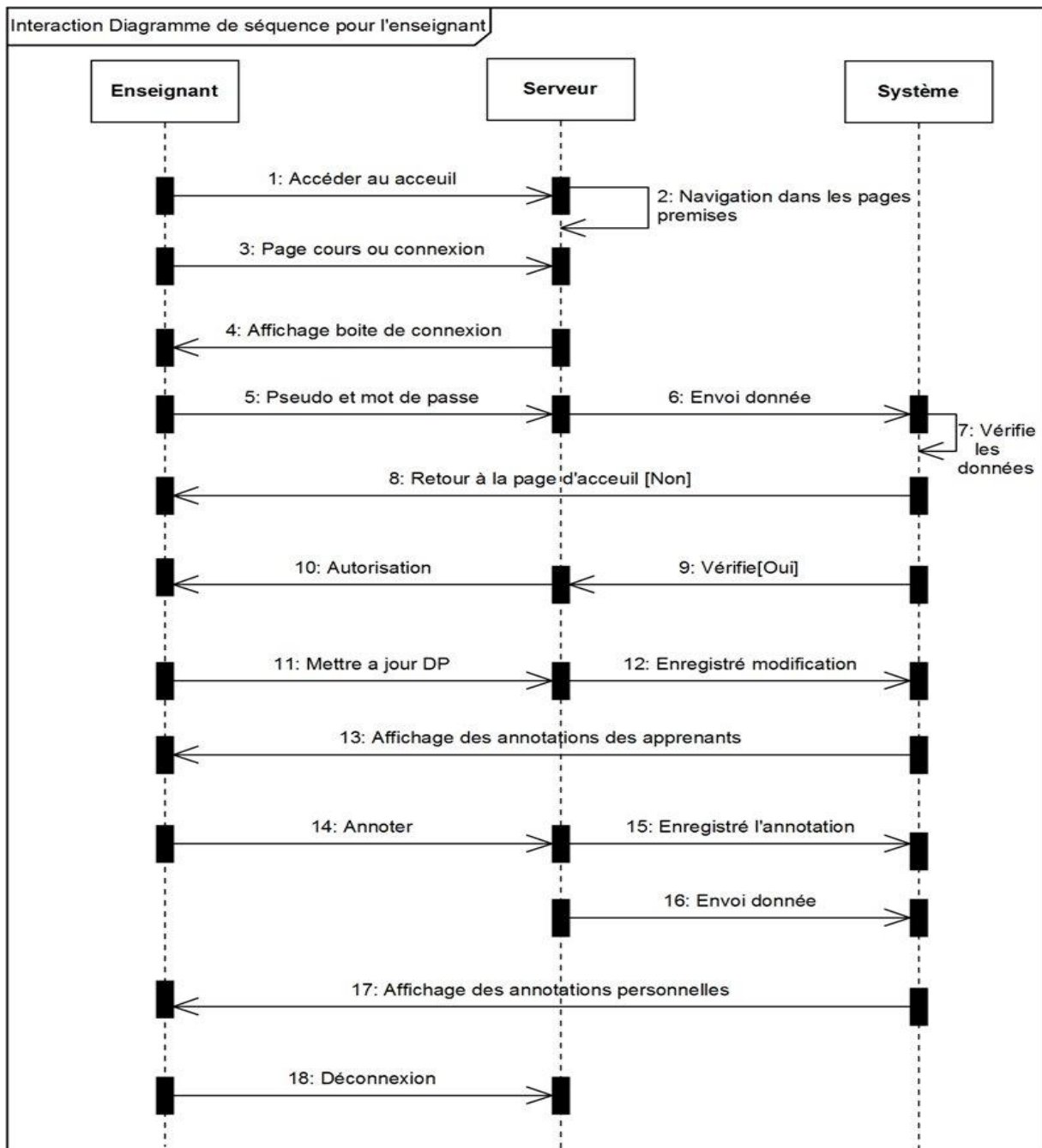


Figure 23 : diagramme de séquence (gestion des enseignants).

3. L'architecteur d'application :

La figure 24 représente l'architecture globale du système. Nous présentons les trois interfaces de notre application, une interface d'enseignant, interface de visiteur et interface d'apprenant.

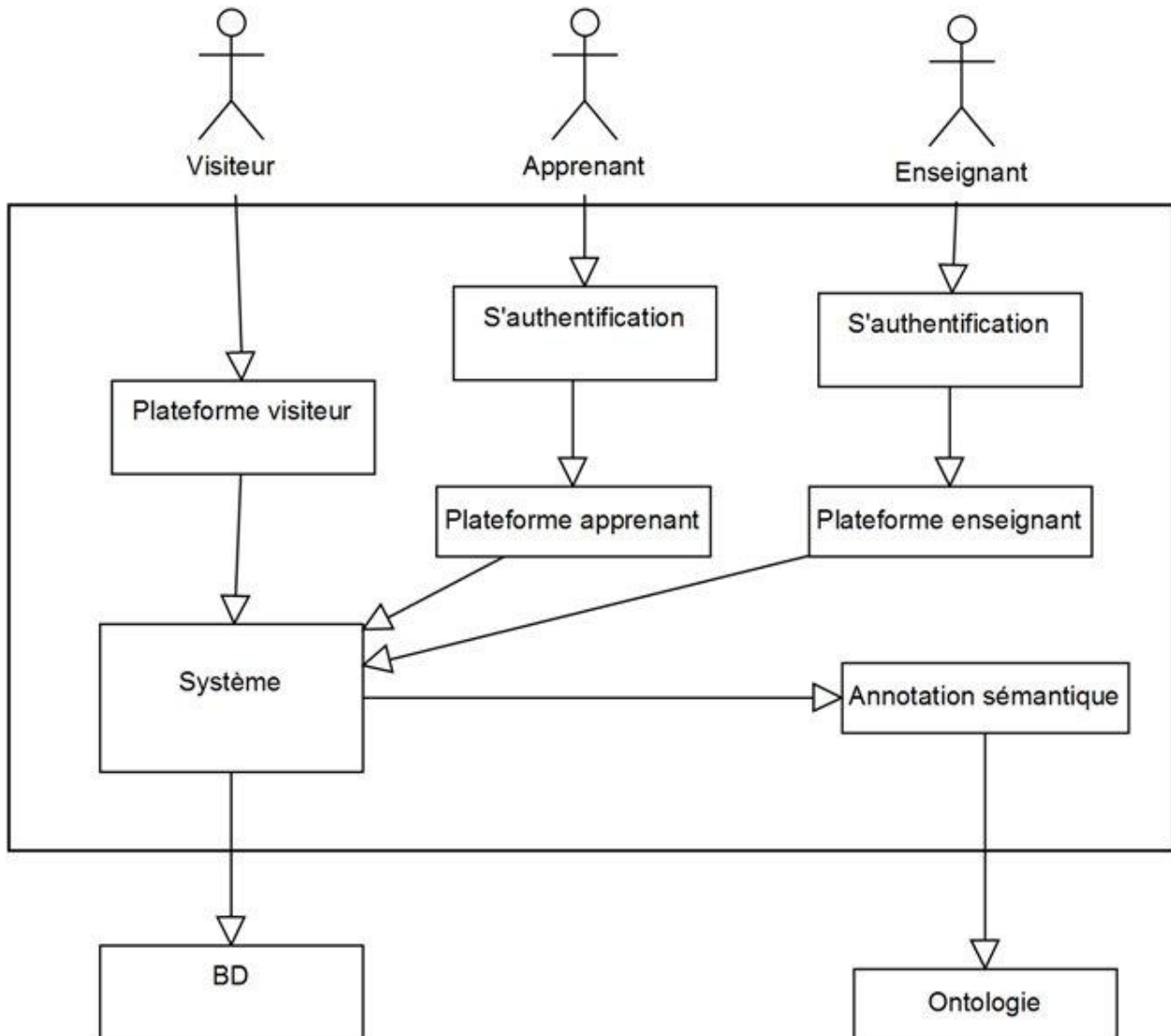


Figure 24: Architecture d'application.

4. Conception de la base de données :

4.1. Modèle conceptuel de données :

Le modèle conceptuel des données (MCD) a pour but de représenter de façon structurée les données qui seront utilisées par le système d'information. Le modèle conceptuel des données décrit la sémantique c'est à dire le sens attaché à ces données et à leurs rapports et non à l'utilisation qui peut en être faite. On établit le MCD après avoir recensé et donné un nom à

l'ensemble des données du domaine étudié. Ensuite on étudie les relations existantes entre ces données (les dépendances fonctionnelles), pour aboutir au MCD.

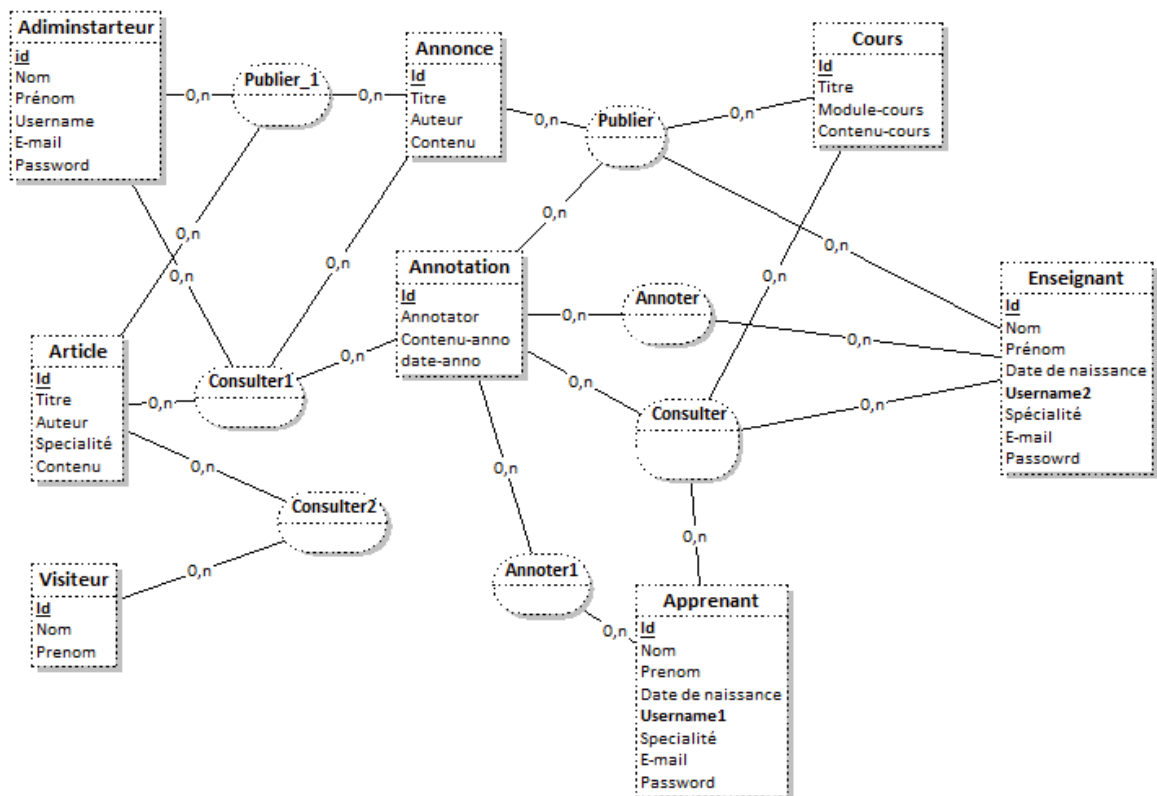


Figure 25 : Gestion de contenu.

4.2. Modèle logique des données :

Le modèle logique présenté ci-dessous a été obtenu en appliquant les règles de passage sur le diagramme de classes.

- ✓ **Administrateur** (Id, nom, prénom, username, email, password)
- ✓ **Visiteur** (Id, nom, prénom)
- ✓ **Enseignant** (Id, Nom, Prénom, Date de naissance, username2, Spécialité, E-mail, password)
- ✓ **Etudiant** (Id, Nom, Prénom, Date de naissance, username1, Spécialité, E-mail, password)
- ✓ **Annonce** (Id, titre, auteur, date, contenu)
- ✓ **Annotation** (Id, Annotator, Contenu-annotation, date)

5. Etape 2 : l'ontologie et l'annotation sémantique

À ce stade, nous expliquons et analysons le cœur de notre projet, qui est des indications récentes à travers l'ontologie, et nous en avons déjà parlé dans le premier chapitre et expliquons ses fondements, et à partir d'ici nous réalisons notre projet, qui est de donner des leçons à l'apprenant le plus facilement possible avec la technologie sémantique et d'utiliser des commentaires sémantiques comme coordinateur de l'étude et demander le système selon la langue Demande SPARQL; Et en le reliant à une question du langage naturel, nous avons donc facilité l'apprentissage des cours par l'apprenant.

5.1. L'ontologie :

L'ontologie du domaine et fonctionnelle et orientée objet, elle est utilisée pour représenter le domaine informatique de cours Algorithme 2 « **Langage C** » sous forme de base de connaissances elle présente les concepts-clés, les attributs, les instances relatifs au domaine.

Les concepts sont organisés dans un graphe dans les relations peuvent être des relations sémantiques et/ou des relations de composition et d'héritage (au sens objet).

Pour réaliser ce type d'ontologie, il existe des éditeurs de structure de base de connaissance le plus utilisé est Protégé. Il est implémenté à partir d'une plateforme Java J2EE, et ils peuvent être exportés dans les formats : RDF(s), OWL et XML schémas.

5.1.1. Les étapes de création d'ontologie :

Lancer Protégé et vous devriez obtenir une configuration semblable à la prochaine figure qui utilise le Look & Feel Protégé qui peut différer légèrement sur différentes plateformes (on peut le modifier via le menu Window). Pour ce tutoriel, vous pouvez ignorer les mises à jour de plugin offertes au lancement du programme. Sur Windows: il faudra peut-être autoriser le lancement d'une application venant d'un développeur non enregistré en passant par les Préférences Système dans la section Sécurité et confidentialité. Définir l'Ontologie IRI pour un URI qui vous ressemble. Dans le menu système File/Save, indiquer le format de sauvegarde de l'ontologie, je suggère RDF/XML et l'endroit sur votre disque où sera conservée l'ontologie.

En principe, votre écran devrait ressembler à celui-ci:

The screenshot shows the Protege ontology editor interface. At the top, the browser address bar displays 'Algo2-Langage-C (http://www.semanticweb.org/zineb/ontologies/2020/1/Algo2-Langage-C)'. Below the browser, the Protege header includes tabs for 'Active ontology', 'Entities', 'Individuals by class', 'DL Query', 'OntoGraf', 'SQWRLTab', and 'SPARQL Query'. The main area is divided into several panels:

- Ontology header:** Shows the 'Ontology IRI' as <http://www.semanticweb.org/zineb/ontologies/2020/1/Algo2-Langage-C> and the 'Ontology Version IRI' as <http://www.semanticweb.org/zineb/ontologies/2020/1/Algo2-Langage-C/1.0.0>.
- Annotations:** A blue bar with a plus sign icon.
- Ontology metrics:** A table showing various counts:

Metrics	
Axiom	590
Logical axiom count	332
Declaration axioms count	214
Class count	139
Object property count	33
Data property count	12
Individual count	32
Annotation Property count	2
Class axioms	
SubClassOf	138
EquivalentClasses	5
DisjointClasses	2
GCI count	0
Hidden GCI Count	10
- Ontology prefixes:** A table listing prefixes and their corresponding URIs:

Prefix	Value
Nombre_d	http://www.semanticweb.org/zineb/ontologies/2020/1/untitled-ontology-7#Nombre_d
langagec	http://www.semanticweb.org/zineb/ontologies/2020/1/untitled-ontology-7#
owl	http://www.w3.org/2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#

At the bottom right, a status bar indicates 'No Reasoner set. Select a reasoner from the Reasoner menu' and 'Show Inferences' is checked.

Figure 26 : Représenté l'écran de protégé.

a) Définition des classes :

Dans l'onglet Entities, choisir le sous-onglet Classes. Cliquer sur le mot owl:Thing dans la vue Class hierarchy qui est la racine de toutes les classes que nous allons créer. En appuyant sur le premier bouton en haut à gauche de cette vue, un dialogue apparaît qui permet de créer une classe qui sera un enfant de la classe sélectionnée. On peut ainsi définir des sous-classes. Le deuxième bouton permet de créer une classe sœur de la classe sélectionnée. Le troisième bouton détruit la classe sélectionnée ainsi que ses sous-classes.

Faire de même pour créer la hiérarchie des classes dont les noms sont en gras dans la liste des énoncés ci-haut pour obtenir une figure semblable à celle-ci (ne pas tenir compte de l'ordre des classes et sous-classes). Ne définir ici que les noms des classes et leur hiérarchie, sans tenir des autres types de contraintes pour le moment. Représenter dans la figure 24.

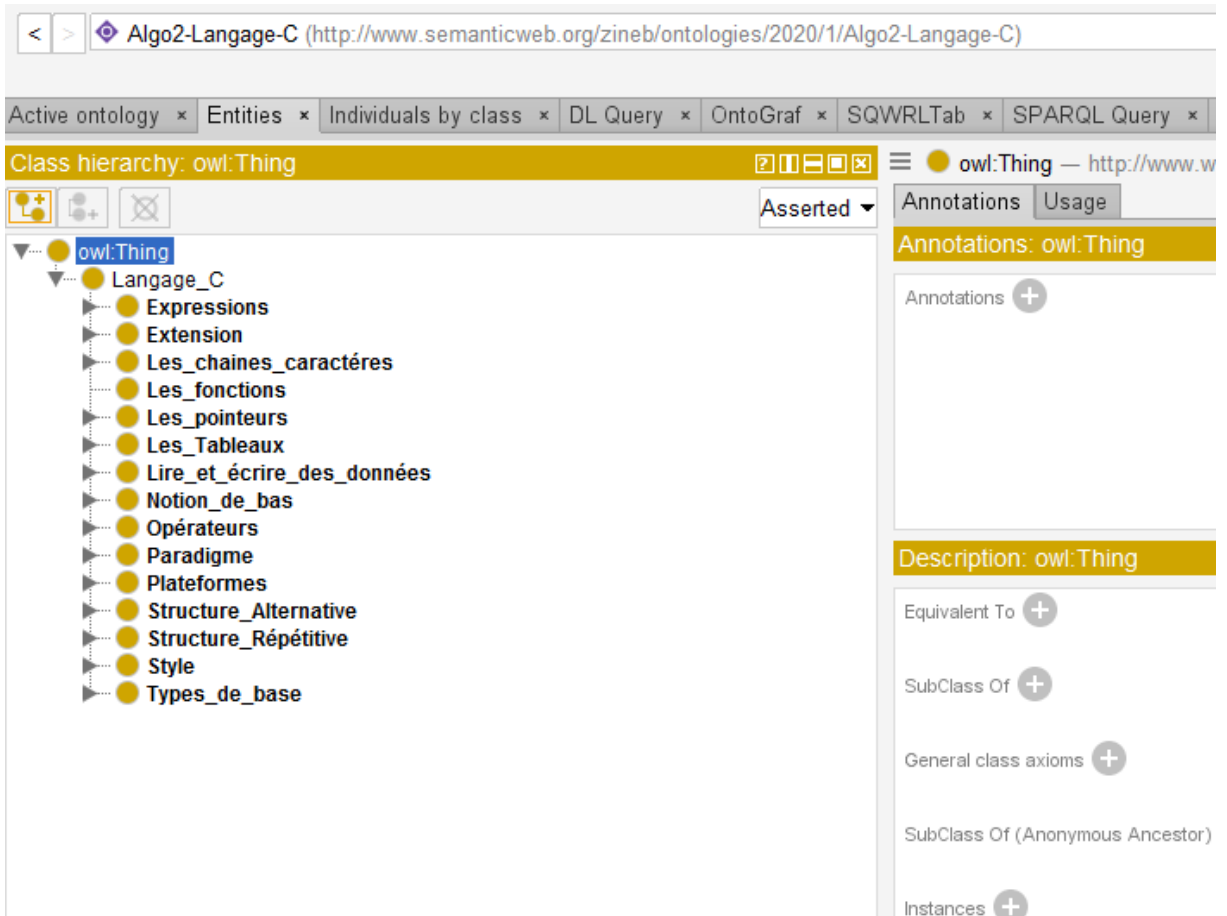


Figure 27 : Représenté les classes d'ontologie.

b) Définition des propriétés :

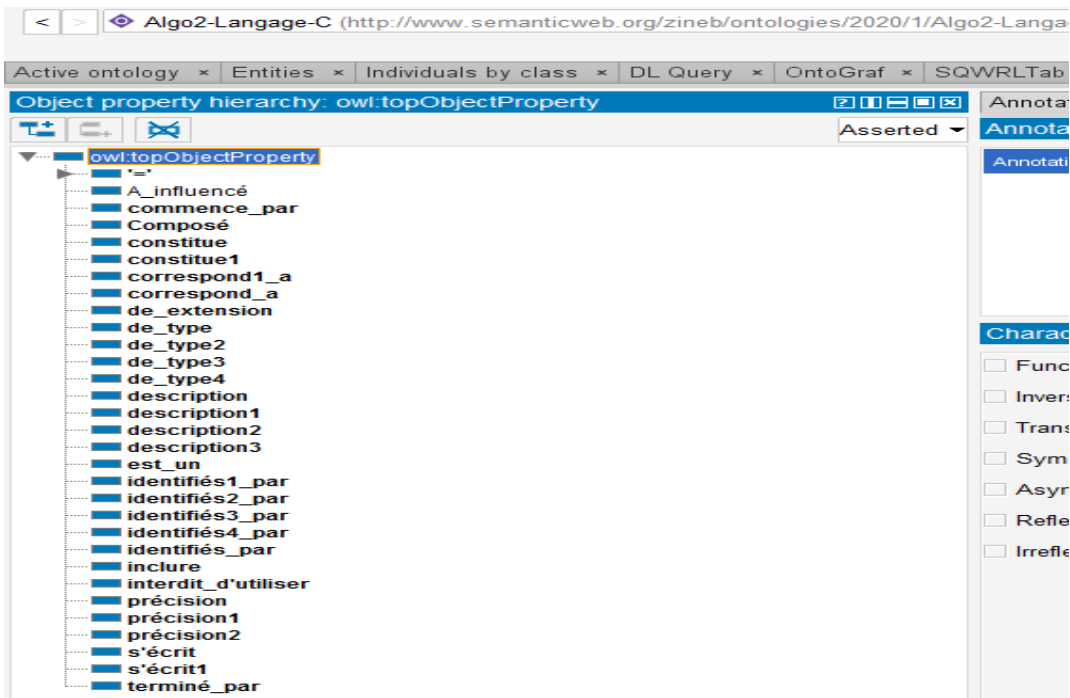


Figure 28 : Représenté les propriétés d'ontologie.

c) Définition des classes complexe :

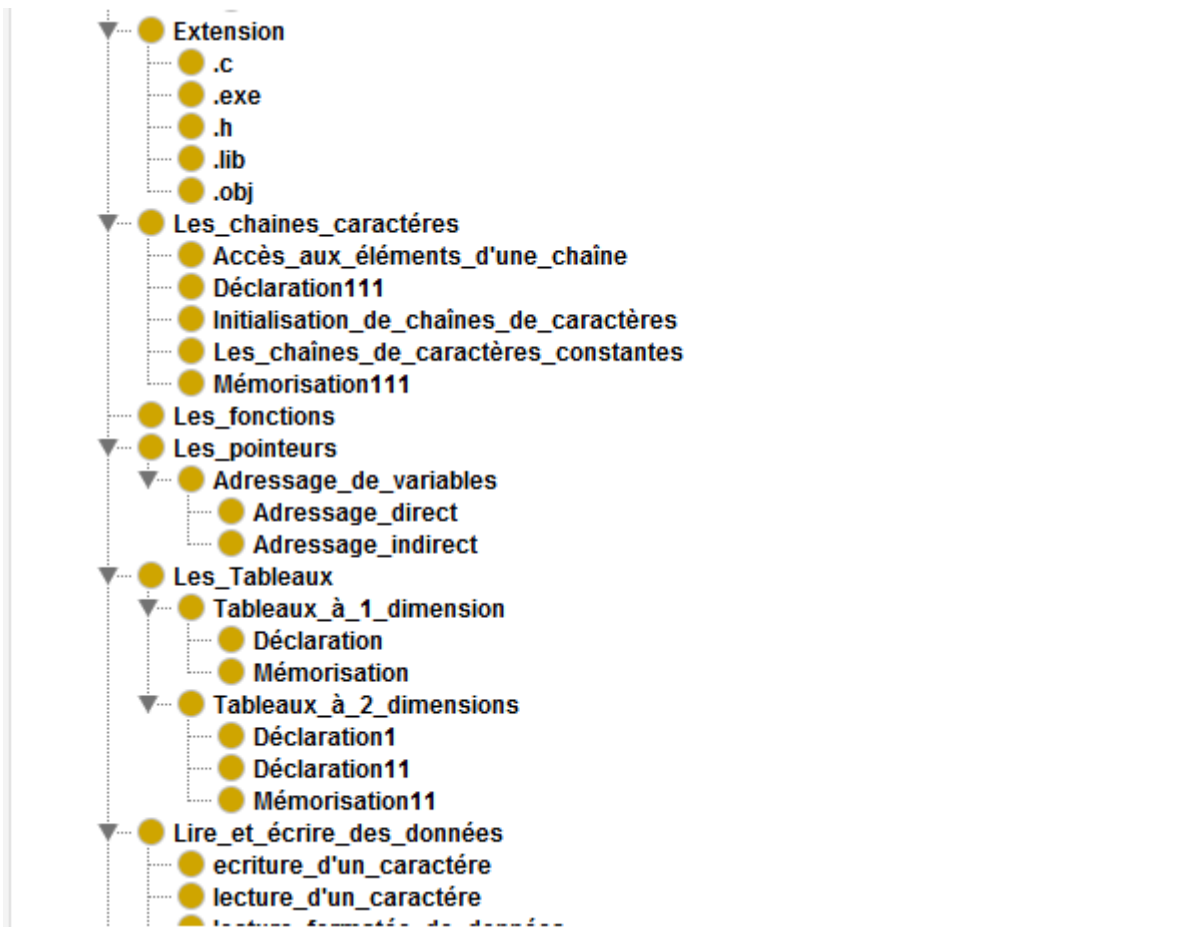


Figure 29 : Représenté une partie des sous-classes.

d) Vérifier de la cohérence d'ontologie :

Le grand intérêt de l'utilisation de Protégé est la possibilité de vérifier si l'ontologie créée ne contient pas des définitions contradictoires. Avec une ontologie aussi simpliste que celle que nous avons construite dans ce document, il est facile de vérifier à la main la cohérence du tout mais ceci est pratiquement impossible pour des cas plus complexes.

Dans le menu Reasoner, sélectionner HermiT, s'il n'y a pas déjà un crochet dans l'item de menu; sélectionner ensuite Start Reasoner.

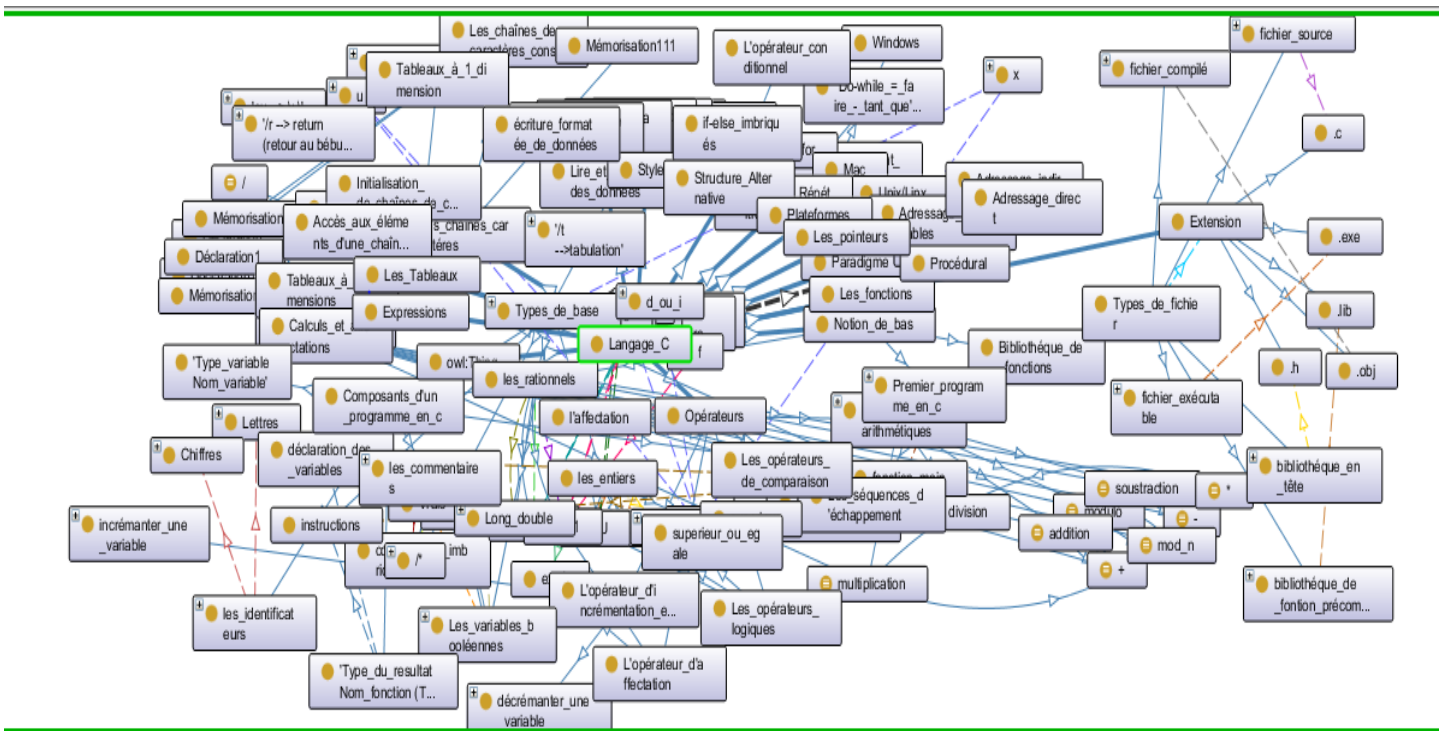


Figure 30 : Représenté le graphe d'ontologie.

5.2. Manipulation d'ontologie :

Le web sémantique est un écosystème complexe qui intègre un ensemble de dispositifs logiciels nécessaires à la modélisation, à la diffusion et à l'exploitation des connaissances entreposées dans une ontologie. En tant qu'environnement de développement intégré, l'IDE Eclipse met à disposition du programmeur Java un ensemble de fonctionnalités apte à cimenter l'écosystème sémantique dans la perspective de construire un environnement de développement d'applications sémantiques. S'adresse à tous les concepteurs intéressés à développer des applications en Java pour le web sémantique. D'une perspective théorique, l'auteur donne un aperçu des principaux usages du web sémantique ainsi que les notions associées à la représentation des connaissances et à l'ontologie. Dans la perspective pragmatique, l'auteur présente une procédure détaillée d'intégration des dispositifs nécessaires au développement d'applications sémantiques en Java ainsi que l'encapsulation en plug-in Eclipse : du Java OWL-API pour la manipulation et la sérialisation d'ontologies, de l'API de Pellet pour assurer le raisonnement logique de l'ontologie, de l'intégration de Protégé pour l'édition d'ontologies, et la configuration d'Eclipse en tant que serveur de développement web d'ontologies. Suivant la séquence de présentation du W3C OWL-2 Primer, chaque élément d'expressivité de la modélisation ontologique est repris, expliqué et schématisé dans le langage G-OWL avant d'être défini en Java dans la syntaxe du Java OWL-API. Une démonstration de raisonnement logique complète la discussion concernant l'expressivité. En tant que fichier informatique, le document ontologique possède des

caractéristiques de traitement qui lui sont propres. Une section du livre est consacrée à la manipulation Java du document ontologique ainsi qu'à la manipulation Java des paramètres du raisonneur.

5.3. Passage de la langue naturelle à une requête SPARQL:

Une tâche difficile dans la réponse aux questions en langage naturel (Q / A pour faire court) sur le graphe de connaissances RDF est de savoir comment combler l'écart entre les questions de langage naturel non structurées (NLQ) et les données RDF structurées en graphe (G). L'un des outils effectives est le «modèle», qui est souvent utilisé dans de nombreux systèmes RDF Q / A existants. Cependant, peu d'entre eux étudient comment générer automatiquement des modèles. À notre connaissance, nous sommes les premiers à proposer une approche de jointure pour la génération de modèles. Étant donné une charge de travail D de requêtes SPARQL et un ensemble N de questions en langage naturel, l'objectif est de trouver des paires q, n , for $q \in D \wedge n \in N$, où la requête SPARQL q est la meilleure correspondance pour la question n en langage naturel n . Ces paires fournissent des conseils prometteurs pour la génération automatique de modèles. En raison de l'ambiguïté des langues naturelles, nous modélisons le problème ci-dessus comme une tâche de jointure de graphe incertaine. Nous proposons plusieurs techniques d'élagage structurel et probabiliste pour accélérer l'assemblage. Des expériences approfondies sur des ensembles de données de référence RDF Q / A réels confirment à la fois l'efficacité et l'efficacité de notre approche. Récemment, les graphiques des connaissances ont attiré beaucoup d'attention à la fois dans le milieu universitaire et dans l'industrie. Le cadre de description des ressources (RDF) étant la norme de facto d'un graphe de connaissances, nous nous concentrons sur le référentiel RDF. Un problème clé est de savoir comment accéder aux graphiques de connaissances et obtenir rapidement les informations souhaitées. Bien que SPARQL soit un langage de requête structurel sur les graphiques RDF, il n'est pas pratique pour les utilisateurs non professionnels d'interroger les graphiques RDF à l'aide de SPARQL, en raison de la complexité de la syntaxe SPARQL et du schéma RDF. Nous illustrons un exemple de requête SPARQL sur DBpedia comme suit.

```
SELECT ?person
WHERE { ?person rdf:type Artist .
        ?person graduatedFrom Harvard University. }
```

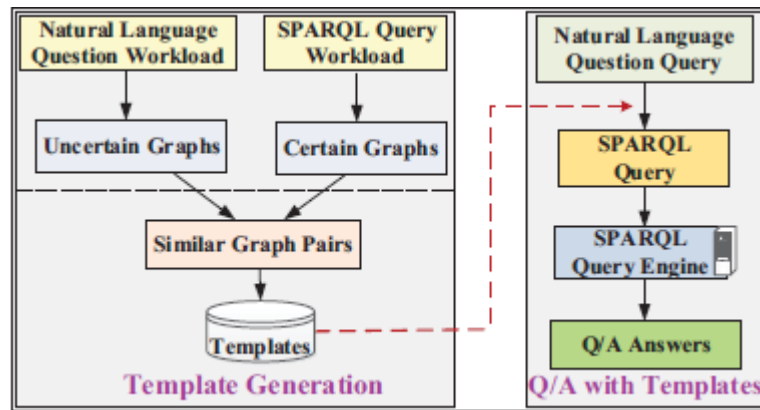


Figure 31 : Cadre pour Q / A RDF basé sur un modèle. [24]

La figure 31 présente le cadre de RDF Q / A à l'aide de modèles. Il se compose de deux tâches: comment générer des modèles et comment utiliser ces modèles dans RDF Q / A. Pour résoudre le premier problème, nous proposons une approche basée sur la charge de travail des requêtes pour la génération automatique de modèles. Plus précisément, les entrées pour la tâche de génération de modèle sont deux charges de travail de requête, L'un est un ensemble D de requêtes SPARQL sur le référentiel RDF (tel que DBpedia workload1). L'autre est un ensemble N de questions en langage naturel, qui peuvent être collectées à partir de certains sites de questions-réponses (Q&A) axés sur la communauté (tels que Yahoo Answers) ou de la charge de travail des requêtes des moteurs de recherche (telles que WebQuestions2). **L'ensemble du processus se déroule en trois étapes comme suit.**

Étape 1. Génération de graphique incertaine. Selon la méthode de, les questions en langage naturel peuvent être interprétées dans des graphes de requêtes sémantiques. En raison de l'ambiguïté de l'interprétation, nous modélisons le graphe de requête sémantique comme un graphe incertain, c'est-à-dire que chaque sommet / arête a plusieurs étiquettes possibles avec différentes probabilités. Ici, les étiquettes sont incertaines en raison de l'ambiguïté sémantique.

Étape 2. Recherche de paires de graphiques similaires. Le but est de trouver certaines paires q, g , for $q \in D \wedge g \in U$, où D correspond aux requêtes SPARQL et U sont les graphiques incertains dérivés des questions en langage naturel³. Ici, nous utilisons la distance d'édition du graphique, qui est largement utilisé pour mesurer la similitude des graphes, pour calculer la similitude entre q et g .

Étape 3. Génération de modèles avec des paires de graphiques similaires. Étant donné une paire retournée q, g , basée sur le mappage entre q et g (le mappage est trouvé lors du calcul de la distance d'édition du graphe à l'étape 2), nous pouvons construire les modèles. Afin de générer des modèles, le problème clé est de savoir comment trouver les paires de graphiques similaires (c'est-à-dire l'étape 2). C'est l'objet de cet article. Nous proposons une nouvelle approche, à

savoir, Similarity Join sur des graphiques incertains, notée SimJ. Plus précisément, nous proposons une série de bornes effectives pour améliorer les performances. Compte tenu des modèles générés, une question en langage naturel est traduite en une requête SPARQL à l'aide des modèles générés lors de la première tâche. Ensuite, la requête SPARQL est recherchée sur les graphiques de connaissances. Algorithme de passage de la langue naturelle à une requête SPARQL :

Ensemble de questions. Nous avons créé A Questions, un ensemble de questions contenant 120 questions sur des données structurées pour vérifier si les systèmes d'assurance qualité existants peuvent bénéficier de notre approche de cartographie adjectivale. Nous avons extrait les 70 questions qui contiennent des contraintes d'adjectif factorisé de QALD-1 à QALD-9, et échantillonné manuellement 50 questions de Yahoo! Réponses. Pour chaque question, une requête SPARQL standard et son résultat d'exécution sur DBpedia-201510 ont également été fournis.

Algorithme 1. Intégration d'Adj2ER dans les systèmes d'assurance qualité existants

Entrée: Une question en langage naturel Q

Sortie: Une requête SPARQL pour la question d'entrée Q

1. procédure GÉNÉRER SPARQL (Q)
2. S := Système de sonorisation existant.GÉNÉRER SPARQL (Q);
3. pour toutes les paires (adj, C) dans Q faire
4. Restrictions := Adj2ER (adj, C);
5. Si S contient une restriction E Restrictions, continuez;
6. S' := S;
7. Supprimez toutes les restrictions dans S' qui ont une similitude sémantique > 0,6 avec adj;
8. pour toute la restriction e Les restrictions faire
9. S* = SPARQL résultant pour ajouter une restriction à S' ;
10. si S* est une requête non vide, alors S: S* ; Pause;
11. Retour S;

5.4. L'annotation sémantique :

L'annotation sémantique consiste à relier le contenu d'un texte à des entités dans une ontologie. Par exemple, pour la phrase «Paris est la capitale de la France.», l'annotation correcte de Paris serait Paris et non Paris Hilton. Elle est variante plus détaillée mais moins exacte de la méthode des entités nommées. De plus en plus fréquemment, on considère la tâche d'annotation sémantique comme l'un des aspects applicatifs du Web sémantique, notamment pour trouver les métadonnées en relation avec l'identité sémantique des données annotées.

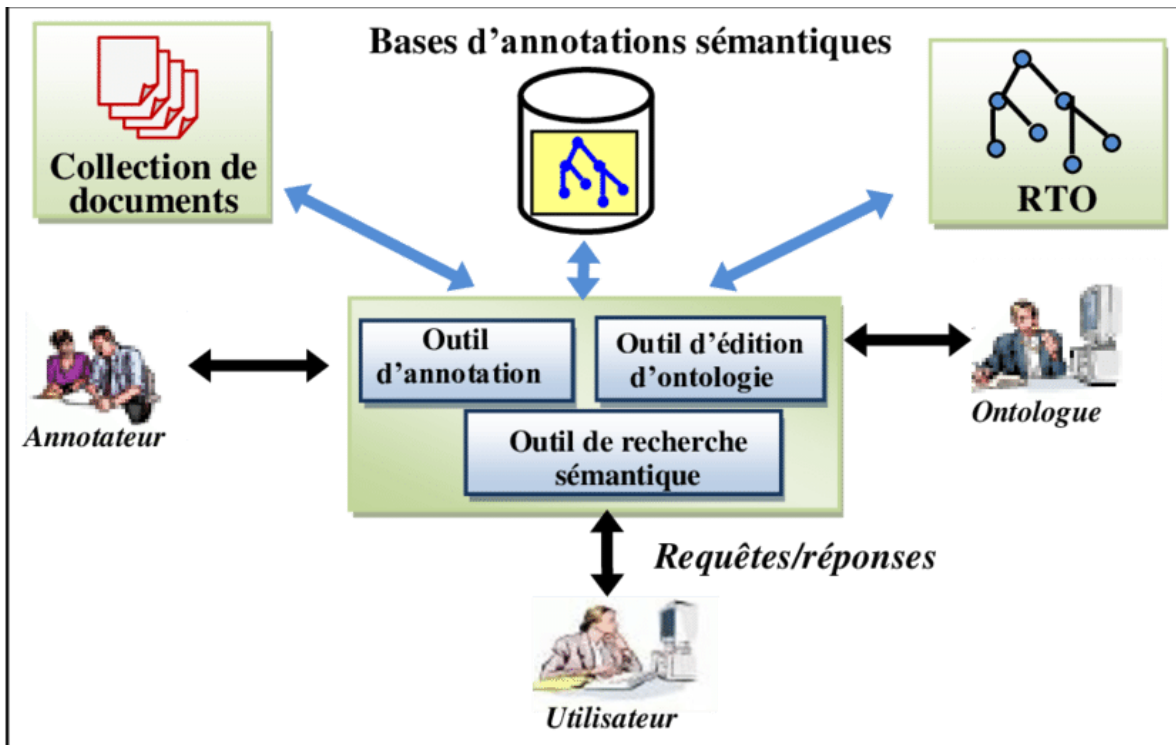


Figure 32 : Architecture de l'annotation sémantique.

5.4.1. Histoire :

La tâche d'annotation sémantique émerge dès les années 1970. [27] comme le principe de mise en relation d'un contenu textuel avec une métadonnée externe qui l'enrichit. Au cours des années suivantes, les méthodes d'extraction d'information vont se focaliser sur des processus plus simples (et à la portée des ordinateurs disponibles) telle que l'extraction d'entités nommées. Mais dès les années 2000, la question de la mise en relation de ces entités avec un contenu extérieur revient à l'ordre du jour par la communauté du traitement automatique du langage naturel. [29]

Plus récemment, l'émergence du web sémantique fait ressortir le besoin d'élaborer des méthodes qui permettraient d'enrichir des annotateurs avec des étiqueteurs sémantiques. Dans le contexte du web sémantique, cette tâche prend alors le nom d'*entity linking* par référence à l'idée d'établir un lien entre un mot ou une séquence de texte et sa représentation sur le réseau Linked Data. [30]

5.4.2. Application :

L'annotation sémantique est une tâche de fouille de texte proche des méthodes de traitement automatique des langues qui consiste à étiqueter dans un document les mots avec des liens qui pointent vers une description sémantique.

Outre l'insertion de contenus qui permettent de compléter un texte (par exemple en identifiant automatiquement les noms de personnes et en fournissant leur date de naissance), l'annotation sémantique joue un rôle essentiel en désambiguïsation en fournissant un identifiant unique représentant parfaitement l'identité du mot ou du groupe de mots annotés.

On peut ainsi définir l'annotation sémantique comme la tâche permettant de déterminer l'identité exacte d'un concept contenu dans un texte et de fournir des informations sur ce concept. [31]

Ses applications sont nombreuses: de la fouille de données intelligente à la conception d'application de compréhension du langage de haut niveau.

5.4.3. Fonctionnement :

Les systèmes d'annotation sémantique recourant à de très volumineuses ressources ontologiques, requises pour la désambiguïsation, ainsi qu'à des modèles de classification complexes et eux aussi parfois volumineux, ils sont le plus souvent livrés sous forme de services Web gratuits ou par abonnement. Certains systèmes tels DBpedia Spotlight sont fournis à la fois sous forme de logiciel serveur et de web service mais sont très complexes à déployer et maintenir. La maintenance est notamment rendue difficile par la nécessité pour ces systèmes de mettre à jour régulièrement leurs ressources de désambiguïsation, par exemple pour couvrir l'apparition de nouveaux concepts.

5.4.4. Méthode standard :

La plupart des méthodes d'extraction et de recherche d'information ont été expérimentées en matière d'annotation sémantique : le modèle vectoriel est le plus répandu dans les systèmes récents, mais des systèmes à entropie maximale ou de simples probabilités conditionnelles donnent également de bons résultats. Le principal obstacle rencontré au cours des trente dernières années pour mettre au point des systèmes d'annotation sémantique résidait dans la disponibilité de ressources libres susceptibles de servir de base de connaissance pour appliquer les algorithmes de calcul de similarité. Désormais, la totalité des systèmes états de l'art (tels que DBpedia Spotlight, l'annotateur de Yahoo! ou encore Wikimeta) exploitent des ressources dérivées des dumps de Wikipédia pour mener à bien le processus de désambiguïsation.

5.4.5. Méthode désambiguïsation:

Le principal problème pour un système d'annotation sémantique réside dans la multiplicité des annotations possibles. Pour annoter une phrase telle que :

Alger est la capitale de l'Algérie.

L'annotateur doit être en mesure de choisir la bonne réponse au sein de la multitude de réponse possibles (il existe une dizaine de villes portant le nom *Alger*, ainsi que des personnes ou encore des navires). Le système d'annotation doit donc mesurer le degré de similarité entre le document et les ressources d'annotation, pour par exemple annoter ainsi en utilisant des URI issues du Web sémantique :

*(Algérie., dbpedia.org/resource/Algérie.) est la capitale de l'
(Alger, dbpedia.org/resource/Alger).*

Généralement, l'annotateur utilise un nombre de n mots situés dans une fenêtre entourant le concept à annoter qu'il stocke dans un vecteur. Ce vecteur est comparé à l'ensemble des vecteurs de mots correspondants extraits d'après la ressource de désambiguïsation (par exemple les dumps de l'encyclopédie Wikipédia). Selon les applications, le calcul de similarité sera réalisé par un produit scalaire pour la similarité cosinus, [32] ou un calcul de probabilité pour une mesure d'entropie. [33]

Ces calculs de similarités produisent une liste de score donné pour chaque entité potentielle, et c'est le rang des candidats dans une liste qui permet de choisir le lien le plus probable.

Le contexte dérivé de Wikipédia utilisé pour réaliser les calculs de similarité peut prendre une forme variable: dans le cas d'ontologie spécialisées pour la désambiguïsation telles que NLGbAse (utilisée par Wikimeta) [34] ou Aleda (utilisée par l'AFP) [35] c'est le texte plein des documents de Wikipédia dont chaque mot est pondéré par un poids TF-IDF qui est retenu. Pour ce qui est de la ressource utilisées par DBpedia Spotlight, les sacs de mots retenus sont constitués de paragraphes entourant l'entité ciblée, dans diverses pages de l'encyclopédie ou l'entité se trouve liée.

5.4.6. Standards de liens:

L'annotation sémantique consiste nécessairement à apposer un descriptif de l'entité ou de la métadonnée correspondante (par exemple une entrée de base de données). Dans un premier temps, des systèmes tels que Kim (Knowledge Integrated Modeling, développé par IBM)¹⁰ ou ceux développés par l'université de Sheffield [36] ont adopté des normes propriétaires.

Désormais, les systèmes - qu'ils soient propriétaires, issus du monde académique ou libres - adoptent en tant que lien sémantique des URI en provenance du Web sémantique (le plus souvent des liens DBpedia, comme pour Yahoo!, Spotlight, Wikimeta, ou encore Alchemy API). Seuls quelques rares systèmes tels Calais (**en**) de Thomson Reuters exploitent une ontologie propriétaire et non compatible avec le Web sémantique.

5.4.7. Problème de l'adaptation linguistique:

Les algorithmes d'annotation sémantique exploitant des ressources de désambiguïsation dérivées de corpus pour fonctionner, ils doivent théoriquement bénéficier d'une ressource pour chaque langue supportée.

Par exemple, un annotateur sémantique qui voudrait apposer des liens sémantiques dans un texte en français doit utiliser une ressource de sacs de mots élaborée depuis un corpus en français (puisque les calculs de similarité entre les différents candidats sont menés d'après le poids de mots).

En pratique très peu d'annotateurs sémantiques sont réellement multilingues. La plupart sont conçus pour l'anglais (Spotlight, Open Calais, Wikimeta) parfois le français (Wikimeta). La majeure partie des systèmes propriétaires présentés comme multilingues ne réalisent pas de désambiguïsation sur les langues autres que l'anglais et produisent en conséquence un très fort taux d'erreur . [37]

5.4.8. Types d'annotation sémantique :

5.4.8.1. Annotation vocale :

L'annotation vocale est une technique informatique qui permet d'analyser la voix humaine captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine.

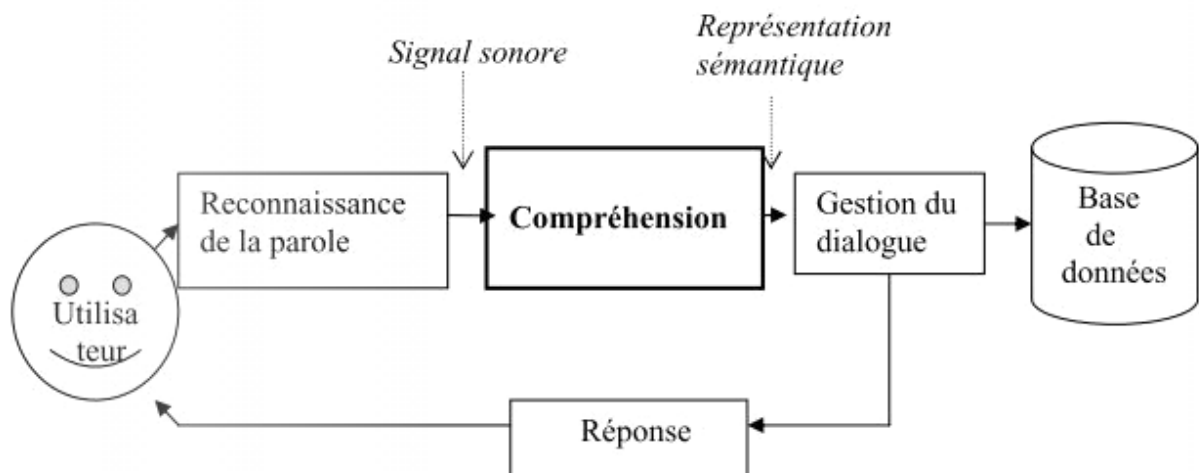
La reconnaissance vocale, ainsi que la synthèse vocale, l'identification du locuteur ou la vérification du locuteur font partie des techniques de traitement de la parole. En particulier, ces technologies permettent de réaliser des interfaces homme-machine (IHM) dans lesquelles une partie de l'interaction se fait au moyen de la voix: «interfaces audio». De même, il est utilisé dans les plateformes d'apprentissage en rencontrant l'apprenant avec sa problématique, le problème

présenté peut être résolu grâce aux web sémantique avec la base de données de la reconnaissance vocale

Dans Parole et dialogue homme-machine, W. Minker et S. Bennacef expliquent que la reconnaissance automatique de la parole est un domaine complexe, car il existe une différence importante entre le langage formel, qui est compris et utilisé par les machines, et le langage naturel, que les humains utilisent. Le langage formel est structuré par des règles syntaxiques strictes et sans ambiguïté. À l'inverse, dans le langage naturel, des mots ou des phrases peuvent avoir plusieurs sens selon l'intonation de l'énonciateur ou le contexte par exemple.

- **Modular Audio Recognition Framework (MARF)** est une plate-forme de recherche et une collection des algorithmes pour le traitement de la voix, le son, le discours, et l'écrit et de langue naturelle (TALN) écrits en Java et arrangé dans un cadre modulaire et extensible qui tente de faciliter l'addition de nouveaux algorithmes. MARF peut servir comme une bibliothèque dans les applications ou est utilisé comme une source pour apprendre et l'extension. Les applications de quelque exemple sont fournies pour montrer comment utiliser le framework. Il y a aussi un manuel détaillé et la référence d'API dans le format de javadoc comme le projet a tendance bien-documenté. MARF et ses applications sont relâchées sous une licence BSD. [38]

Grâce aux progrès récents de la reconnaissance vocale, les systèmes de Dialogues Oraux Homme/Machine (DOHM) deviennent progressivement accessibles au grand public. La plupart de ces systèmes permettent d'interroger une base de données et leur architecture générale correspond au schéma suivant:



Malgré des avancées significatives, les systèmes de DOHM ne sont actuellement destinés qu'à des tâches extrêmement précises interrogation sur les horaires d'avion ou de train, informations météo, etc. Par ailleurs, le dialogue y est généralement dirigé par la machine et peu d'initiatives sont laissés leurs utilisateurs qui doivent souvent se contenter de répondre aux questions qui leur sont posées. Pour réaliser des systèmes permettant des interrogations dans des domaines un peu plus larges ou un dialogue plus naturel, des progrès importants sont nécessaires. Améliorer la compréhension automatique est l'un de ces nécessitent

Après avoir donné un aperçu rapide des problèmes posés par la compréhension automatique et

de ce que la linguistique de corpus peut apporter dans ce domaine, nous décrivons dans cet article un projet ministériel d'évaluation de la compréhension, pour lequel a été recueilli et annoté un corpus de dialogue oral ; nous essayons ensuite de circonscrire et préciser les questions soulevées par cette expérience. [39]

5.4.8.2. Annotation image :

L'annotation automatique d'images est le procédé par lequel un système informatique assigne automatiquement une légende ou des mots clés à une image numérique. Cette application des techniques issues de la vision par ordinateur est utilisée dans les systèmes de recherche d'images pour organiser et retrouver des images d'intérêts dans une base de données.

Cette méthode peut être considérée comme un type de classification d'images multi-classe avec un très grand nombre de classes - de la taille du vocabulaire utilisé. Typiquement, l'image est d'abord analysée par l'extraction de vecteurs de descripteurs, et les mots pour l'apprentissage de l'annotation sont utilisés avec des techniques d'apprentissage automatique pour pouvoir être capable d'affecter automatiquement des annotations à de nouvelles images. Les premières méthodes dans le domaine apprenaient les corrélations entre les descripteurs d'images et les annotations, puis des techniques utilisant des méthodes issues de la traduction automatique ont vu le jour, permettant de faire une traduction entre le vocabulaire textuel, et ce qu'on considérait comme un vocabulaire visuel, où chaque mot serait un groupe de régions similaires, obtenues par regroupement (*clustering*). D'autres travaux utilisent par exemple des approches de classification ou des modèles de pertinences.

Les avantages de l'annotation automatique d'images par rapport à la recherche d'image par le contenu (CBIR) sont que les requêtes peuvent être spécifiées plus naturellement par l'utilisateur, sous forme de requête textuelle. Dans les systèmes de CBIR, l'utilisateur doit rechercher les images par des concepts tels que la couleur ou la texture, ou alors doit donner une image exemple pour laquelle il souhaite trouver des images similaires. Les méthodes traditionnelles de recherche d'images, telles que celles utilisées par les documentalistes, reposent fortement sur les annotations manuelles d'images, ce qui est très coûteux en temps et en argent, et devient impossible étant donné la taille et la croissance des bases de données d'images actuellement existantes.

5.4.8.3. Annotation vidéo :

Une vidéo annotée est une vidéo contenant des annotations (des sous-titres). Elle dépasse la limite d'une vidéo qu'on visionne de bout en bout, car elle transmet de l'information supplémentaire telle des commentaires, des questions, des remarques.

La vidéo peut-être utilisée pour soutenir l'apprentissage et l'enseignement dans plusieurs domaines. La vidéo numérique permet de stocker et de rendre disponible de grandes quantités de documents audiovisuels. La vidéo permet aussi de parcourir le contenu de manière non linéaire, ce qui facilite grandement la sélection et le visionnement de segments vidéo, et de combiner rapidement des segments vidéo issus de sources différentes. L'autre élément qui peut être associé à la vidéo est donc la possibilité d'annoter de manière automatique ou manuelle le contenu visuel

ou sonore d'une vidéo et d'amalgamer ces annotations à la vidéo pour créer de nouveaux documents multimédias. De grandes quantités d'information peuvent ainsi enrichir le contenu audiovisuel. L'ajout d'annotations à la vidéo ouvre toutes sortes de possibilités sur le plan du développement de ressources médiatiques pour soutenir l'apprentissage et l'enseignement.

L'utilisation de vidéos annotées présente un vrai intérêt pédagogique, surtout dans des séquences pédagogiques axées sur des comportements et des savoir-faire comme des gestes techniques.

Voici quelques utilisations possibles :

- Tutoriel d'utilisation de logiciels (=vidéographie /screencast)
- Filmer des gestes
- Filmer des participants à une activité sociale (ex entretien d'embauche)
- Démonstrations didactique
- Enseignement de langue (dialogue)
- Analyse de mouvements
- Témoignages
- Vidéo pour donner du feedback
- Vidéo d'une animation

6. Conclusion

Dans ce chapitre, nous nous sommes concentrés sur les deux étapes, la première étape concernant les aspects analytiques et conceptuels de notre application à travers le diagramme UML , l'architecture du système, le MCD et MLD, et l'étape 2 est liée à la façon de définir l'ontologie, de traiter l'ontologie, le passage du langage naturel à la requête SPARQL et enfin de définir le commentaire. Explicatif sémantique en général, compte tenu de son importance dans le domaine de l'apprentissage et de l'acquisition des connaissances.

Le chapitre suivant sera consacré à l'implémentation et à la présentation de notre Application.

Chapitre 3 : Implémentation et Réalisation de l'application

Dans cette partie, nous allons présenter les langages et le choix des outils dans un premier temps, puis nous continuerons avec des étapes de l'implémentation et de réalisation et nous terminerons avec une présentation de l'application finale.

1. Environnement de développement:

1.1. L'environnement Hardware :

- **Ordinateur** : ASUS X540S.
 - ✓ **Système d'exploitation** : Windows 8.1 64 Bits.
 - ✓ **Processeur** : Intel Dual-Core N30 50, up to 2.16 GHZ.
 - ✓ **Mémoire installée (RAM)** : 4GO (296 GO utilisable).
 - ✓ **Type du système** : Système d'exploitation 64 bits, processeur X64.

1.2. L'environnement Software :

- **PaceStar UML :**



PaceStar UML est un environnement de modélisation, prenant en charge une large gamme de modèles et de diagrammes, et fournit des fonctions d'assistance modèle et de contrôle de cohérence.

- **Notepad++ :**



Notepad++ est un éditeur de texte libre générique, fonctionnant sous Windows, codé en C++, qui intègre la syntaxique de code source pour les langages et fichiers C, C++, Java, C#, XML, HTML, PHP, JavaScript, makefile, art ASCII, .bat, MS

fichier, ASP, VisualBasic/VBScript, SQL, Objective_C, CSS, Pascal, Perl, Python, R, M ATLAB, Lua, TCL, Assembleur, Ruby, Lisp, Scheme, Propriétés, Smalltalk, PostScript et VHDL ainsi que pour tout autre langage informatique, car ce logiciel propose la possibilité de créer ses propres colorations syntaxiques pour un langage quelconque. Ce logiciel est également une alternative au bloc-notes de Windows.

- **MySQL :**



MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle et Microsoft SQL Server.

- **phpMyAdmin (PMA) :**



PhpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL.

- **WampServer :**



WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP..WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un icône de la barre (icône près de l'horloge de Windows). La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

- **Protégé 3.4.1 :**



Protégé est un système auteur pour la création d'ontologies. Il a été créé à l'université Stanford et est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique. Protégé est développé en Java. Il est gratuit et son code source est publié sous une licence libre (la Mozilla Public License). Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, etc.

- **Npm :**



npm est le gestionnaire de paquets officiel de Node.js. Depuis la version 0.6.3 de Node.js, npm fait partie de l'environnement et est donc automatiquement installé par défaut. npm fonctionne avec un terminal et gère les dépendances pour une application. Il permet également d'installer des applications Node.js disponibles sur le dépôt npm. En avril 2020, GitHub annonce l'acquisition de npm.

1. Les langages Utilisés :

- **UML :**



Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation

graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

- **Html :**



L'HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). HTML est initialement dérivé du Standard Generalized Markup Language (SGML).

- **Php :**



Hypertext Preprocessor, plus connu sous son sigle *PHP* (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook , Wikipédia , etc. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

- **JavaScript:**



JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript dédié pour l'interpréter, indépendamment des considérations de sécurité qui peuvent se poser le cas échéant.

C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet de créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel. JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances npm, avec environ 500 000 paquets en août 2017.

JavaScript a été créé en 1995 par Brendan Eich. Il a été standardisé sous le nom d'ECMAScript en juin 1997 par Ecma International dans le standard ECMA-262. Le standard ECMA-262 en est actuellement à sa 8^e édition. JavaScript n'est depuis qu'une implémentation d'ECMAScript, celle mise en œuvre par la fondation Mozilla. L'implémentation d'ECMAScript par Microsoft (dans Internet Explorer jusqu'à sa version 9) se nomme JScript, tandis que celle d'Adobe Systems se nomme ActionScript.

JavaScript est aussi employé pour les serveurs avec l'utilisation (par exemple) de Node.js ou de Deno.

- **Bootstrap:**



Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

- **XML:**



Le langage XML est un langage qui permet de décrire des données à l'aide de balises et de règles que l'on peut personnaliser. Pour Extensible Markup Language, désigne un langage informatique (ou métalangage pour être plus précis) utilisé, entre autres, dans la conception des sites Web et pour faciliter les échanges d'informations sur Internet. Il s'agit, en quelque sorte, d'une version améliorée du langage HTML avec la définition de balises supplémentaires.

- **RDF :**



RDF est le langage de base du Web sémantique. L'une des syntaxes (ou sérialisations) de ce langage est RDF/XML. D'autres syntaxes de RDF sont apparues ensuite, cherchant à rendre la lecture plus compréhensible.

- **SPARQL:**



SPARQL est un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles à travers Internet. Son nom est un acronyme récursif qui signifie SPARQL Protocole and RDF Query Language.

2. Présentation et Test de l'application :

Notre application représente la mise en œuvre d'une plate-forme d'apprentissage en ligne de l'université Chadli Bendjdid El-tarf (UCBET) et le système d'annotation (LINKED DATA) d'un document pédagogique basé sur Linked Data.

2.1. Interface de visiteurs :

Lorsque vous consultez à la première notre Plate-forme vous affichez une page d'accueil principale de l'université rassemble plusieurs liens et fonctionnalités, et vous donnez certains réagis limités tant que vous êtes visiteur.

Dans la figure 33 prise d'écran de la page d'accueil.

NOS SERVICES

Nous offrons la possibilité de se préparer à la vie



Etudiants

Cher étudiant préférez pour assister et téléchargez vos leçons.

ENTRER



Enseignants

cher Enseignant poster votre Cours, Leçons, Articles.

ENTRER



Livres

bibliothèque électronique Livres disponibles.

ENTRER

DERNIÈRES NOUVELLES

Recevez les dernières nouvelles et les meilleurs articles aujourd'hui



La plateforme de formation à distance

29 SEPT 2020 ADMIN

Bienvenue à vous sur la plateforme de formation à distance de l'Université Chadli Bendjedid El-Tarf (UCBET). Cette plateforme vise à mettre à disposition de la communauté universitaire un espace d'échange collaboratif...



Linked Data

29 SEPT 2020 ADMIN

Le Web des données (linked data, en anglais) est une initiative du W3C (Consortium World Wide Web) visant à favoriser la publication de données structurées sur le Web, non pas sous la forme de silos de données isolés les uns des autres, mais en les reliant entre elles pour constituer un réseau global d'informations...

Figure 33 : Page d'accueil de la plate-forme LINKED DATA.

Dans la figure 34 prise d'écran de la page Contacter l'administration :

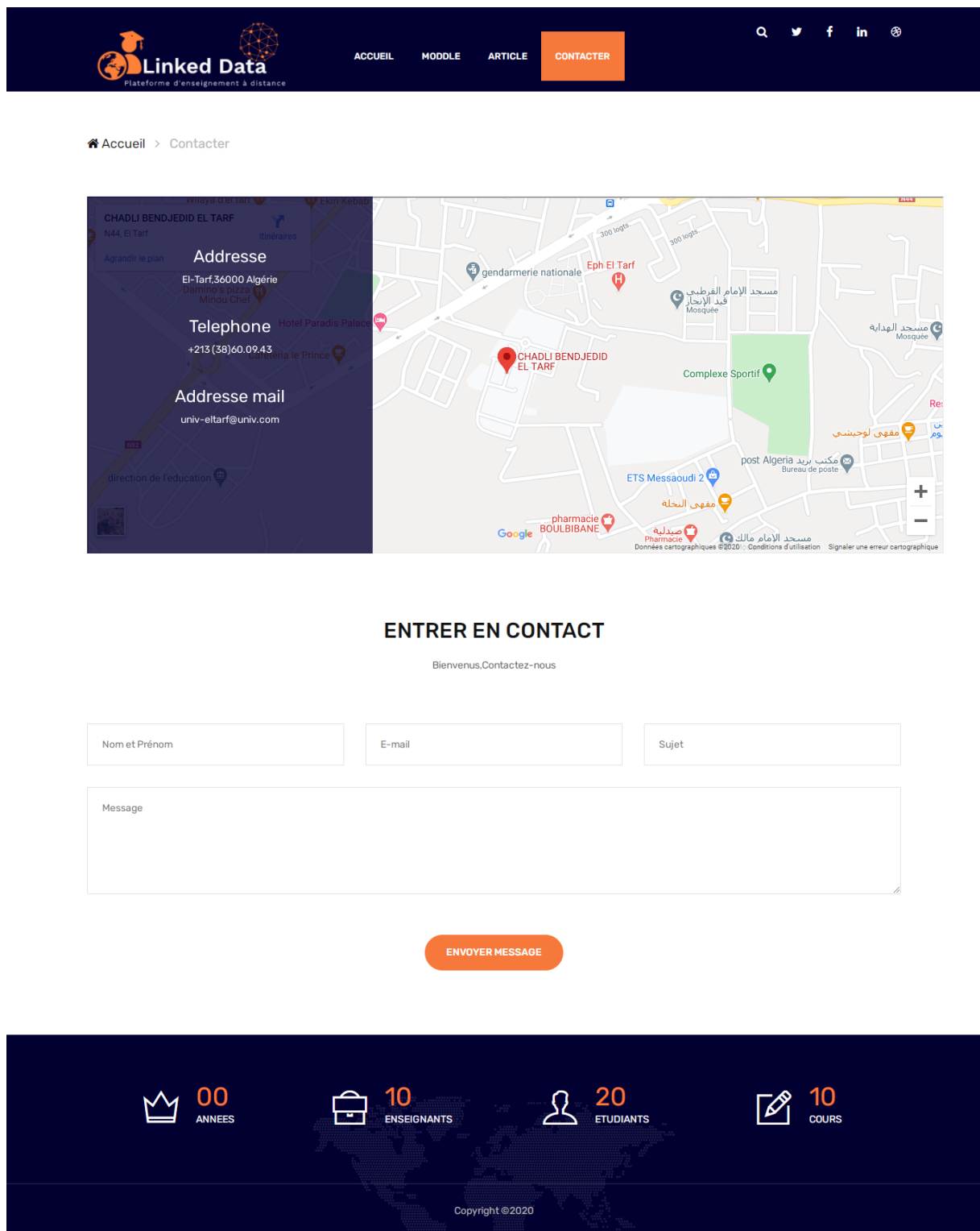


Figure 34 : Page de Contacter.

Dans la figure 35 prise d'écran de la page de publication de l'article ; L'objectif de cette partie de la plateforme est aux chercheurs de placer leurs articles de recherche au profit et de développer la recherche.

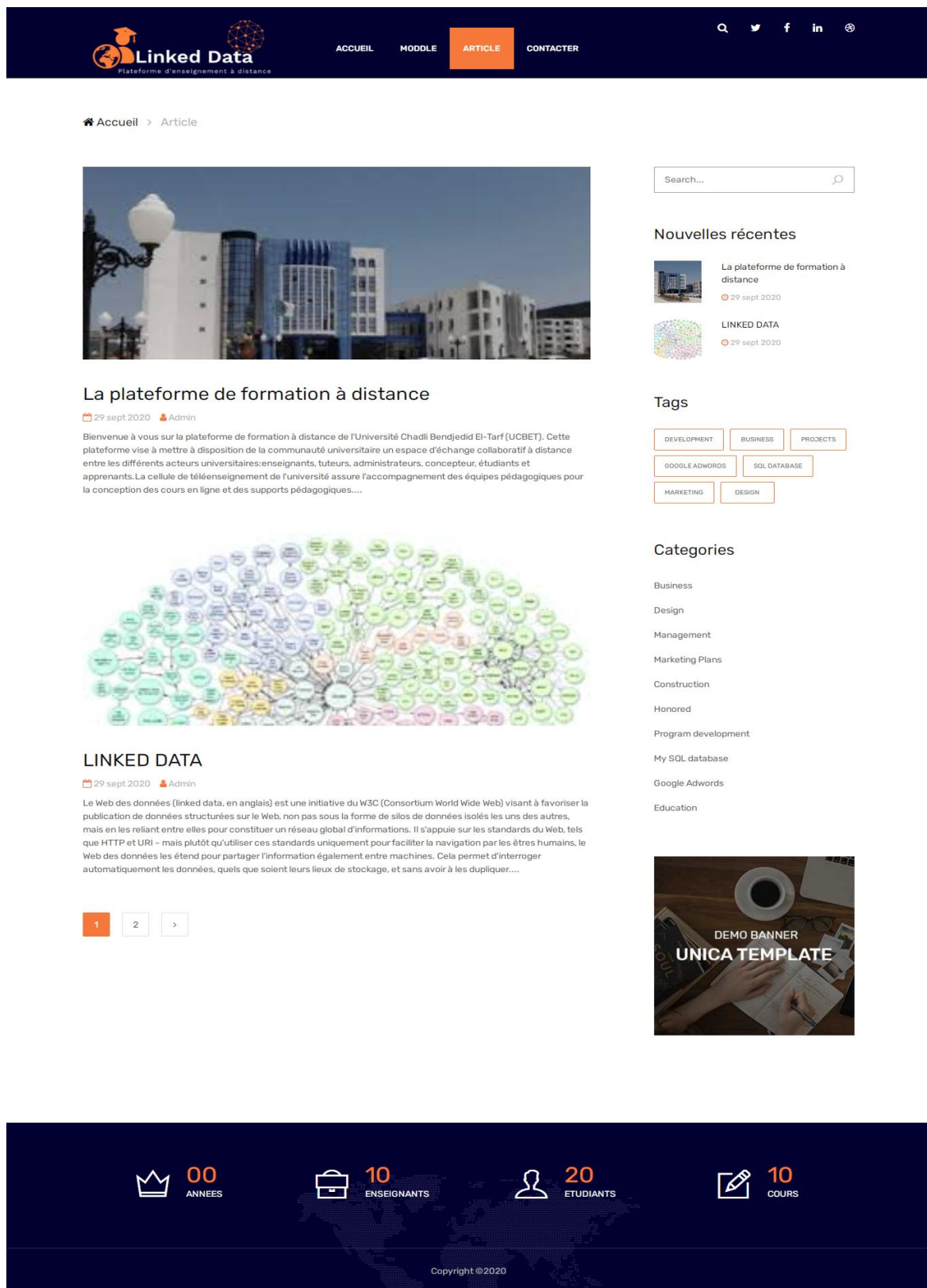


Figure 35 : Page de Consultation des articles.

2.2. Interface d'utilisateurs :

Dans cette description vous été soit un Enseignant ou Apprenant. On a réservé un espace à chaque une, donc il y a fort possible balayage globale de site bien déterminé et une exploitation des différents fonctionnalités que vous donne la plate-forme de façon mieux que visiteur en résumé ça comme suit :

2.2.1. Espace d'Enseignant :

En présentant quelque fonctionnalité du la plate-forme de l'espace d'enseignant :

Dans la figure 36 prise d'écran de la page de connexion sur la plateforme.

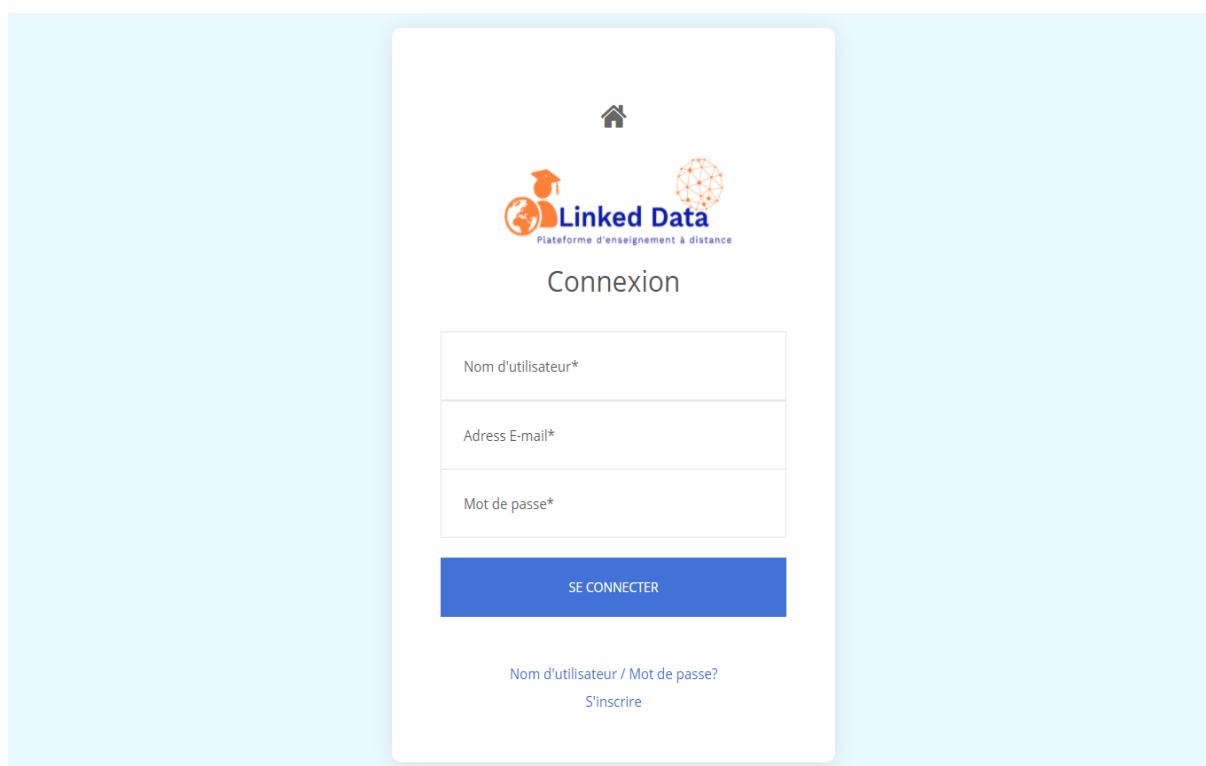


Figure 36 : Page de Connexion d'enseignant.

Dans la figure 37 prise d'écran de la page de services d'enseignants.

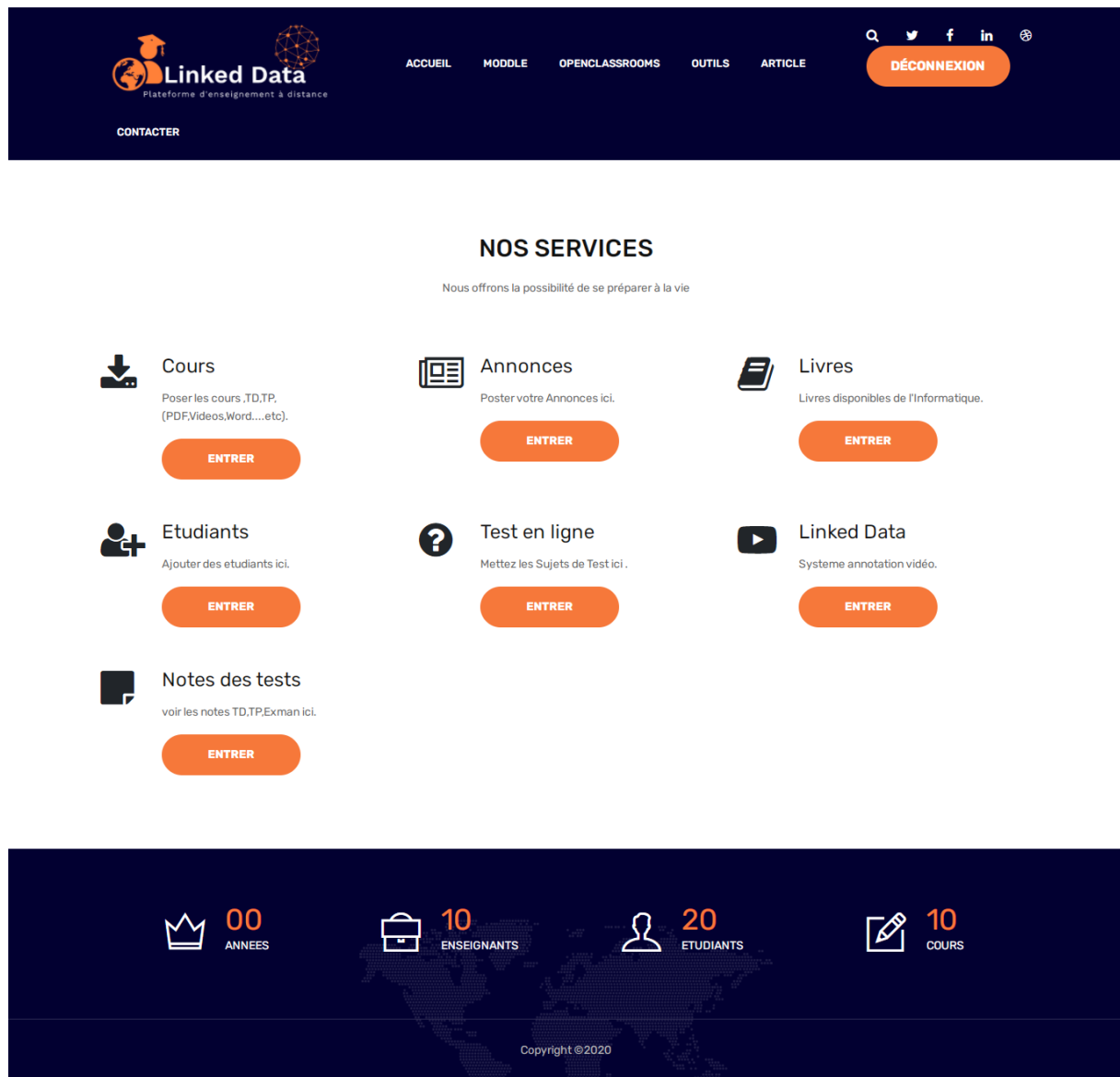


Figure 37 : Page d'accueil espace d'enseignant.

Dans la figure 38 prise d'écran de la page de système d'annotations et consulter les annotations des apprenants.

The screenshot displays the 'Systeme Vidéo annotation' interface. At the top, a dark blue navigation bar contains the 'Linked Data' logo and menu items: 'ACCEUIL', 'COURS', 'MODDELE', 'OPENCLASSROOMS', 'OUTILS', 'ENSEIGNANT', and 'DÉCONNEXION'. The main heading is 'Systeme Vidéo annotation', with a subtitle: 'Facilite l'absorption et la compréhension des cours par chaque apprenant à l'aide d'une technique vidéo annotation basé sur Linked data'. Below this, a video player is shown with the title 'Introduction To C Language'. The video content lists four topics: 'Introduction', 'REASONS FOR POPULARITY OF C', 'USES OF C', and 'First Program in C'. Above the video player are buttons for 'Ajou vidéo', '+ Ajouter A', '+ Linked data', and a download icon. To the right, an 'Annotations' sidebar lists four comments from users, each with a 'Reply' button and a 'No likes' indicator. At the bottom, a dark blue footer bar features three orange boxes with icons and text: '30 Apprenants', '10 Cours', and '0 Années complètes'. A small copyright notice 'All Rights Reserved. © 2020 SmartEDU Design By: html design' is visible at the very bottom.

Figure 38 : Page de système d'annotation (espace d'enseignant).

2.2.2. Espace d'apprenant :

En présentant quelque fonctionnalité de la plate-forme de l'espace d'apprenant :

Dans la figure 39 prise d'écran de la page Connexion pour les apprenants ; l'objectif est de permettre aux apprenants d'accéder à la plateforme et de bénéficier d'assister à leurs cours et autres activités à distance.

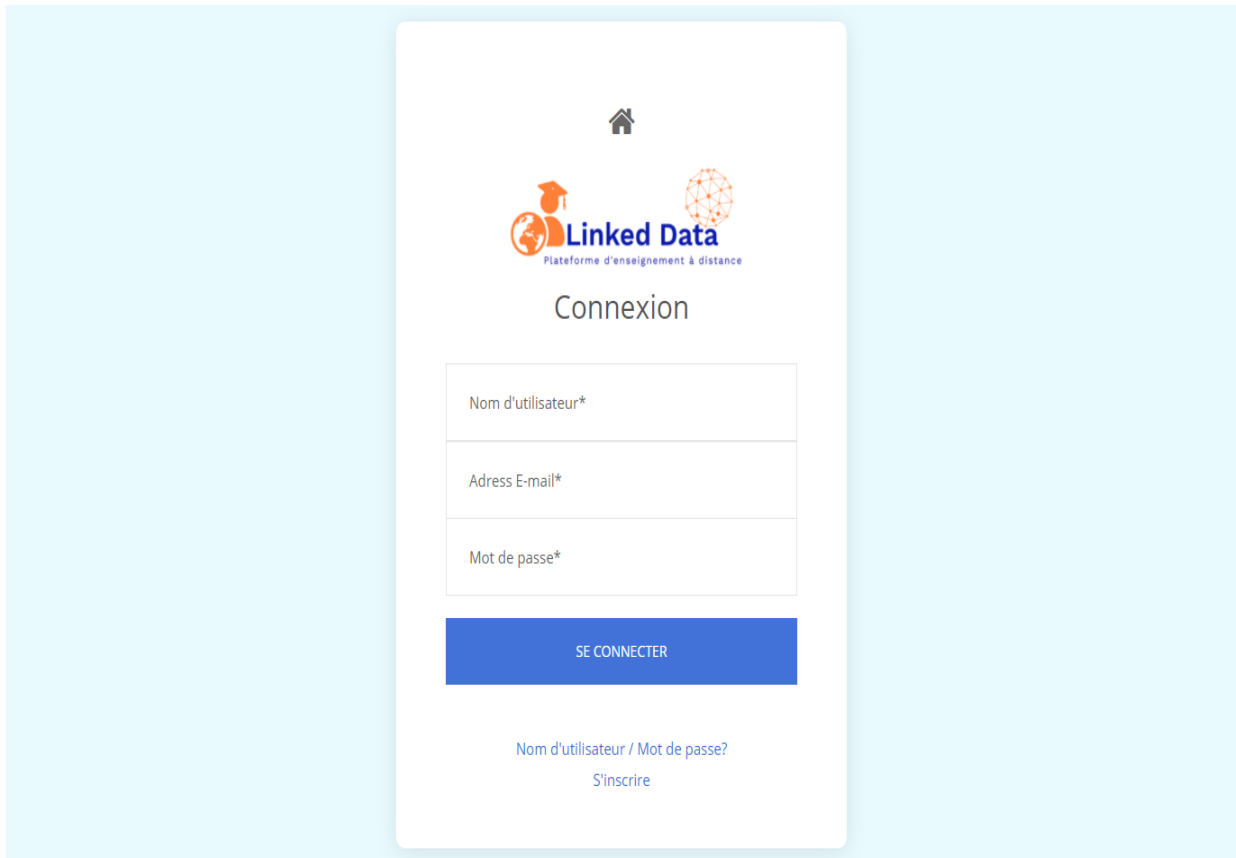


Figure 39 : Page de Connexion d'Apprenant.

Dans la figure 40 prise d'écran de la page de services pour les apprenants ; l'objectif est de bénéficier d'assister à leurs cours et autres activités à distance.

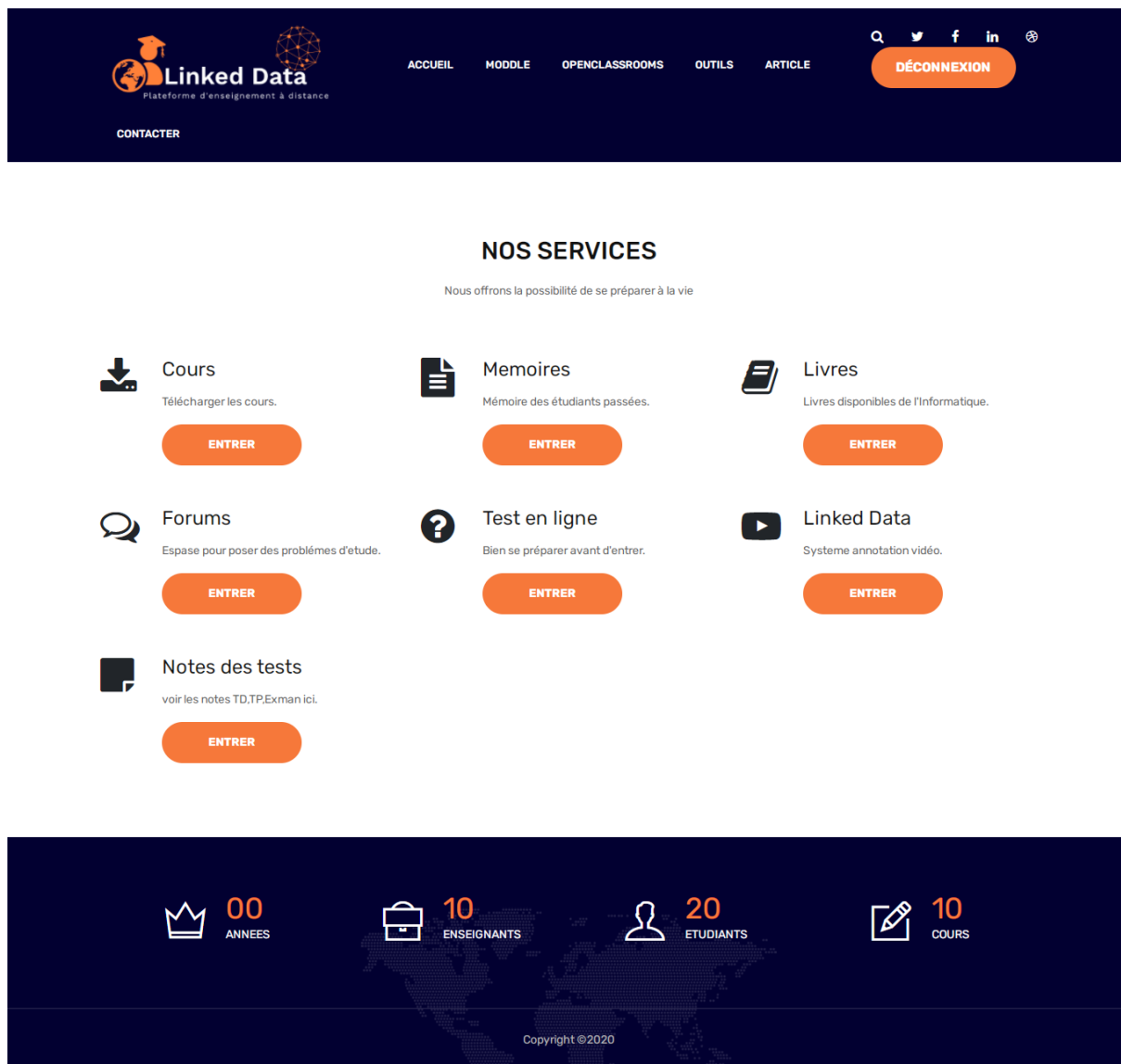


Figure 40: Page d'accueil d'apprenant.

Dans la figure 41 prise d'écran de la page de Linked data pour les apprenants ; l'objectif est assister et annoter à leurs cours à distance.

Systeme Vidéo annotation

Facilite l'absorption et la compréhension des cours par chaque apprenant à l'aide d'une technique vidéo annotation basé sur Linked data

Annotations

- de Apprenant à 2020-09-28 18:43:38
2.07-2.10.Waht is operating system in c?
Reply
No likes
- de Systeme à 2020-09-28 18:52:58
un logiciel système qui gère le matériel informatique.
Reply
1 likes
- de Apprenant à 2020-10-11 15:04:32
Dennis ?
Reply
No likes
- de enseignant à 2020-10-11 15:12:28
????
Reply
No likes
- de Apprenant à 2020-10-12 14:20:49
?????
Reply
No likes

Quiz

Lorsque vous avez terminé d'assister à le cours,laissez votre réponse ici

- Quiz 1: Introduction Language C
- Quiz 2: Structure de base du programme C - Programmation C
- Quiz 3: Fonction
- Quiz 4: Condition et tableaux

30 Apprenants | 10 Cours | 0 Années complètes

Figure 41 : Page de système d'annotation (espace d'apprenant).

Dans la figure 42 prise d'écran de la page de Linked data pour les apprenants ; comment ajouter une annotation dans une vidéo.

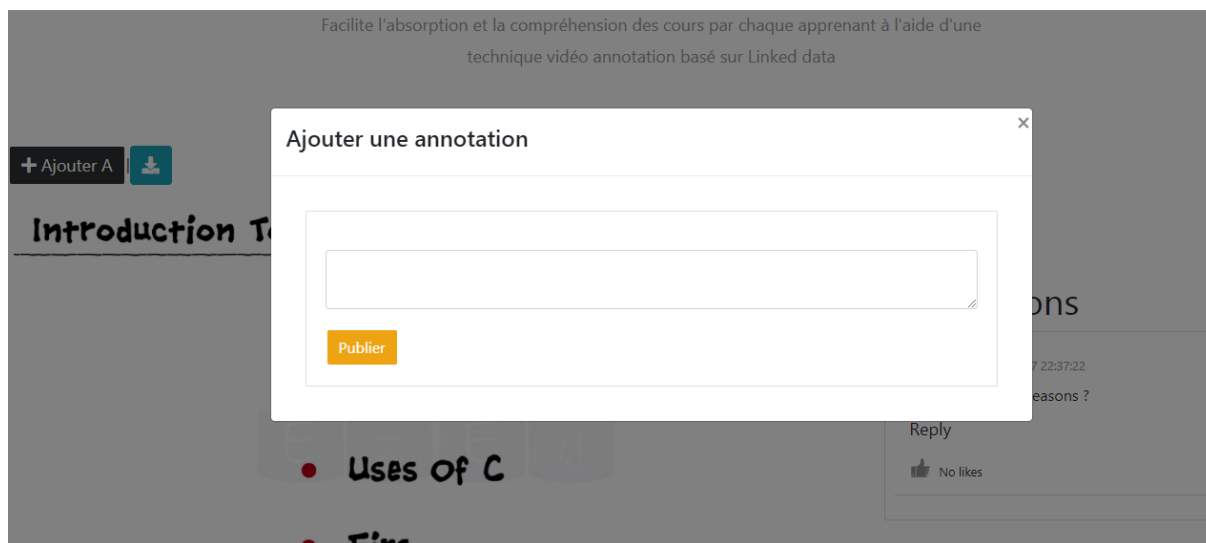


Figure 42 : Page de système d'annotation (création d'annotation).

Dans la figure 43 prise d'écran de la page de Linked data en utilisant le système Spotlight pour les apprenants; comment ajouter une annotation dans une vidéo, donne la réponse exacte.

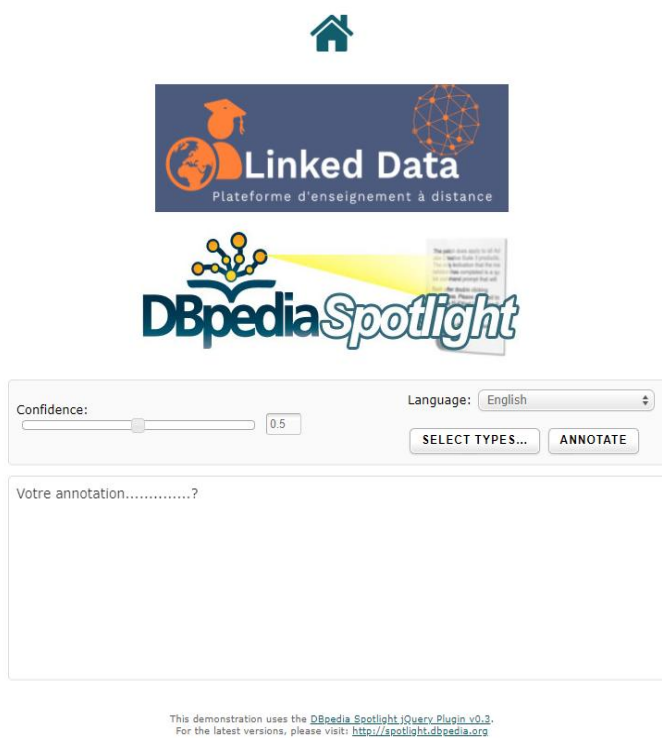


Figure 43 : Page de système d'annotation Spotlight (création d'annotation).

Dans la figure 44 prise d'écran de la page de connexion pour les apprenants ; pour le questionnaire en ligne, l'apprenant entre ses informations et débute le quizz et que terminé le système vous envoie leur note et la réponse. Enfin le système envoie leur Quizz à l'email de l'enseignant.

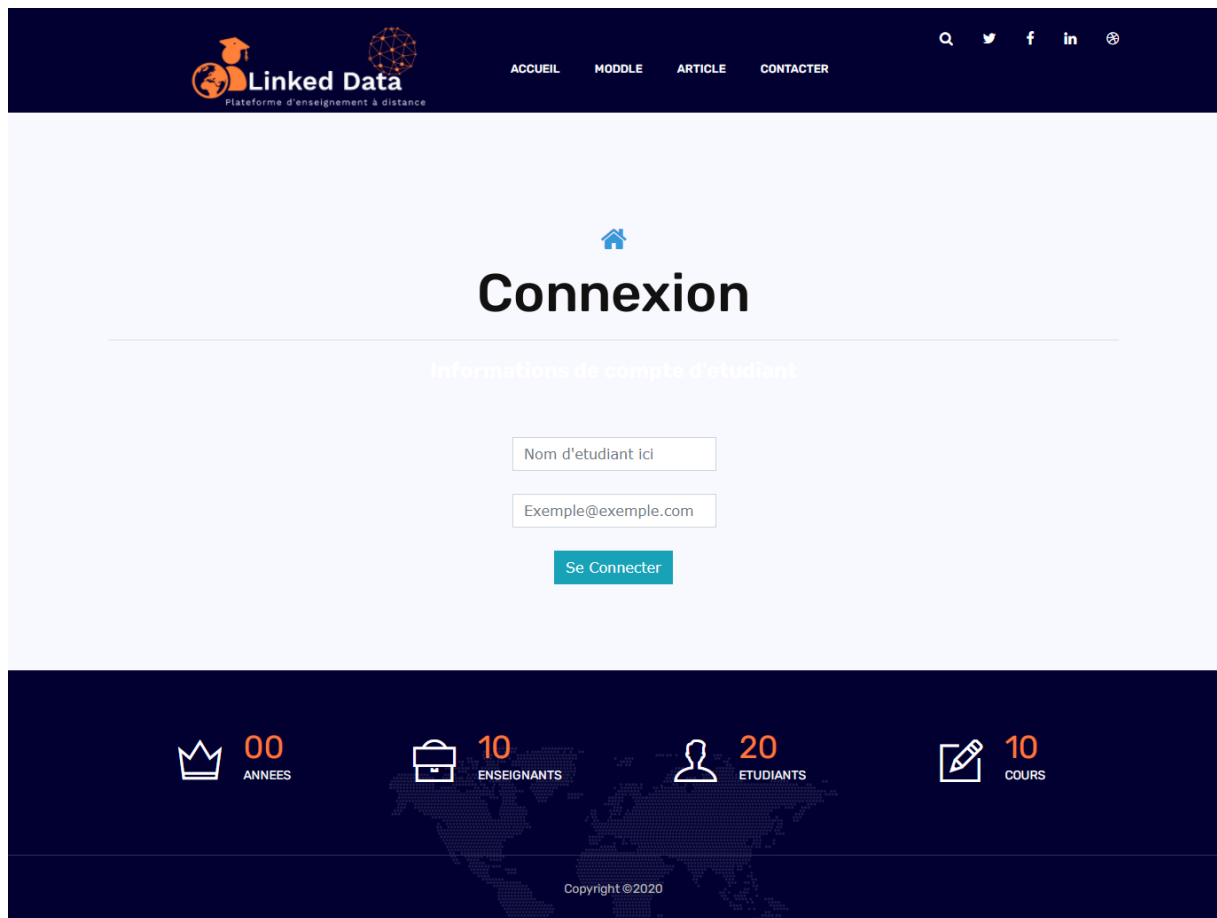


Figure 44 : Page de Quizz (login de l'apprenant).

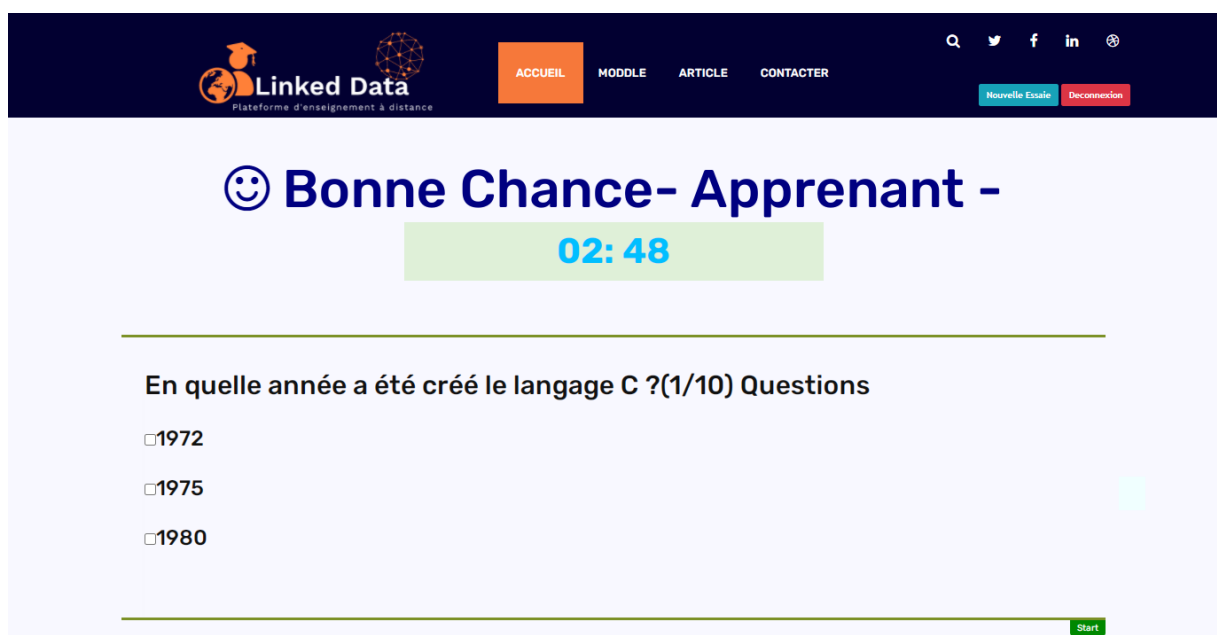


Figure 45 : Page de Quizz de l'apprenant.

2.2.3. Espace d'administrateur :

En présentant quelque fonctionnalité de la plate-forme de l'espace d'administrateur :

Dans la figure 46 prise d'écran de la page de connexion pour les administrateurs.

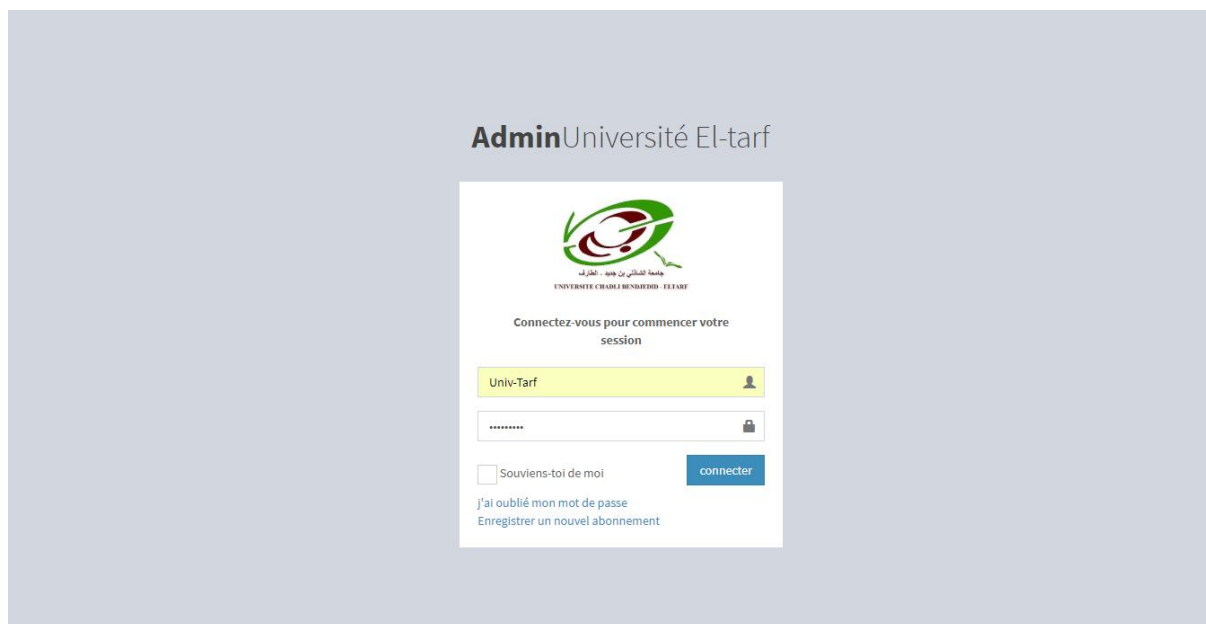


Figure 46 : Page Connexion d'administrateur.

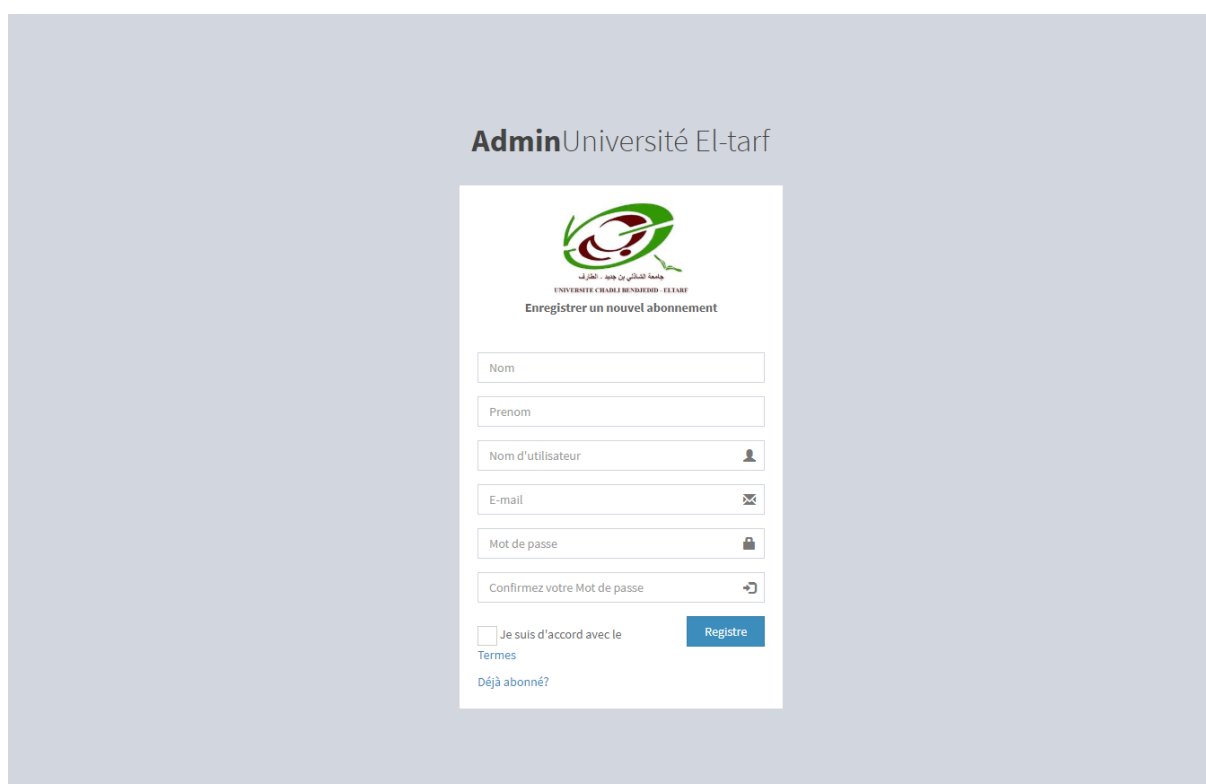


Figure 47 : Page de registre d'administrateur.

Dans la figure 48 prise d'écran de la page d'accueil pour l'administration ; pour consulter les utilisateurs et les activités de la plateforme.

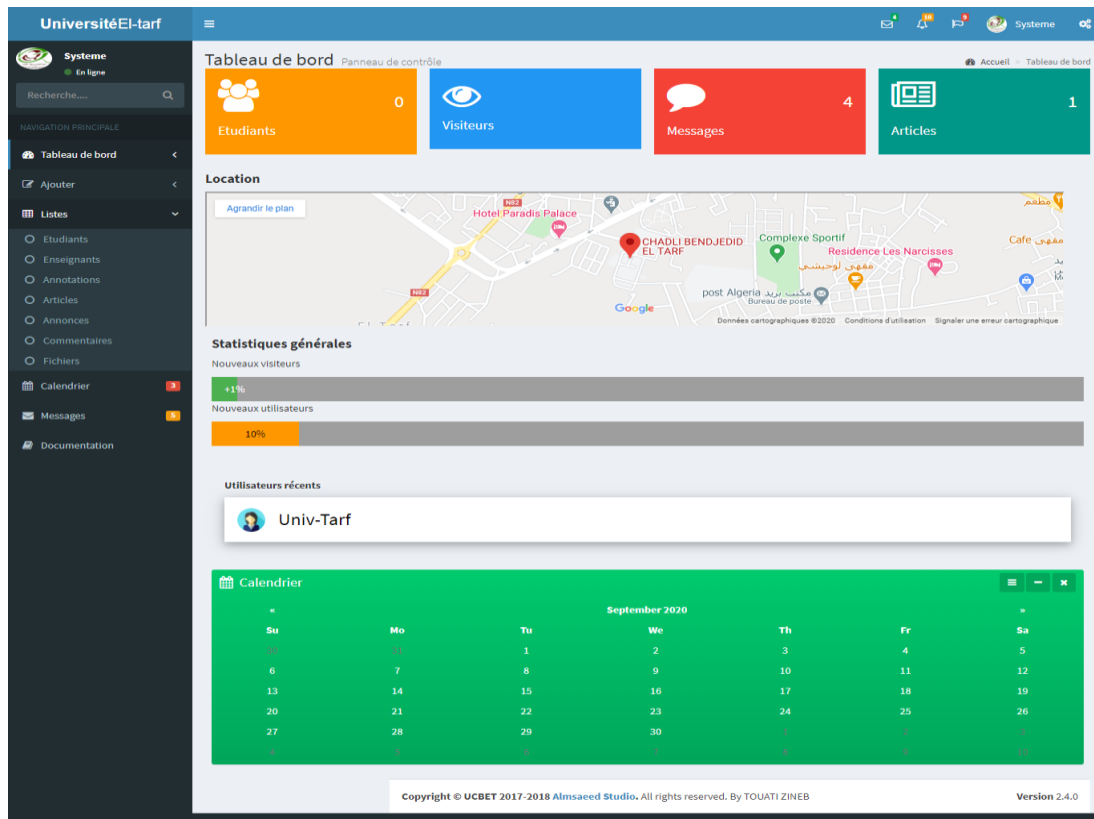


Figure 48 : Page accueil d'administrateur.

Dans la figure 49 prise d'écran de la page de consultation de système d'annotation et il (modifier et supprimer) pour l'administrateur.

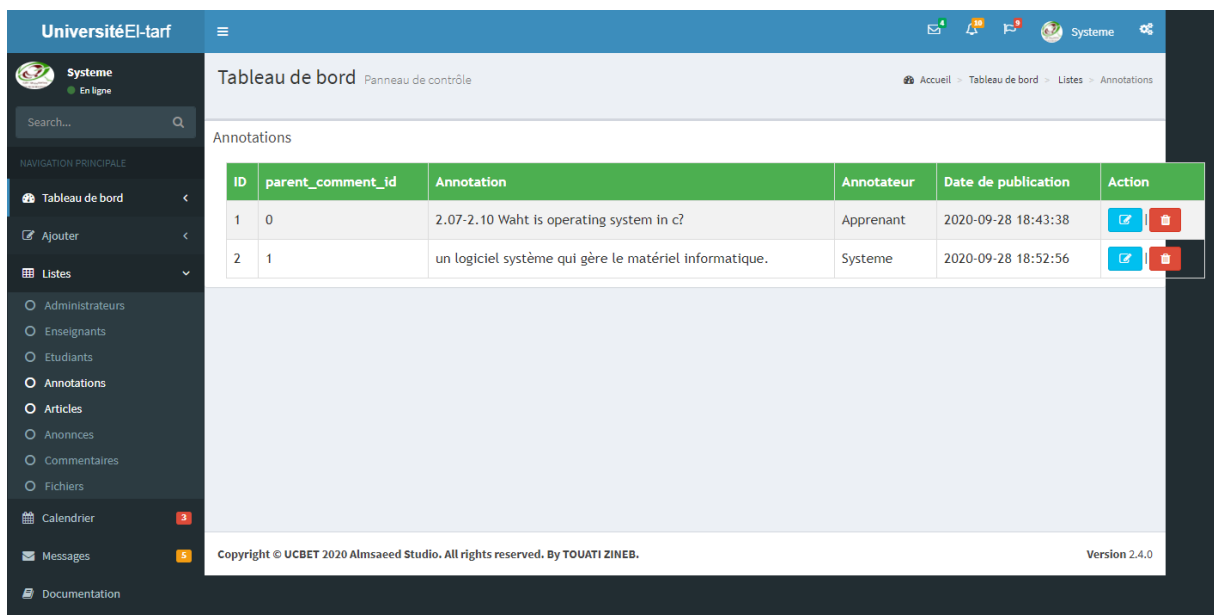


Figure 49 : Page affichage des annotations d'administrateur.

3. Conclusion:

Nous avons vu dans ce chapitre les outils utilisés pour le développement, puis les étapes de réalisation et d'implémentation, ensuite nous avons montré à travers quelques interfaces une vision globale sur le fonctionnement de notre système. Enfin avec ce chapitre nous terminons la phase de développement de ce système.

Conclusion et Perspectives

Le développement fulgurant des technologies de l'information, des réseaux et des moyens de communication est à l'origine d'une expansion prodigieuse de l'internet. Nous traversons une nouvelle ère où la démocratisation de l'internet et le passage à la société de l'information conduisent à un processus de globalisation incitant à la mise en relation des personnes et à la constitution d'équipes virtuelles réparties à travers le monde. Ainsi l'ère des plates-formes collaboratives construites dans des domaines différents (gestion des connaissances, enseignement à distance, Télémédecine, etc.), intègrent plusieurs outils de collaboration (synchrones et asynchrones) et offrent des possibilités d'interaction et d'échange d'informations entre des groupes de personnes situés dans des endroits différents.

Les travaux développés dans ce mémoire s'inscrivent dans le domaine des plates-formes d'enseignement en ligne collaboratives intégrant les différents outils de collaboration (synchrones et asynchrones).

Les perspectives de notre travail consistent à :

- ❖ Etendre la plateforme vers une implémentation d'un système intelligent.
- ❖ Procéder au processus d'hébergement du site web et de son accessibilité en ligne.
- ❖ Enrichir la plateforme avec système l'annotation d'un document pédagogie et un ensemble des autres outils de collaboration comme wiki et la Visioconférence ...etc.
- ❖ Optimiser la plateforme pour faciliter sa maintenance et son extension future.
- ❖ Assurer la sécurité de notre plateforme

- [1] Philippe L., Chantal R., Jean C. (2002) “ Sur quelques aspects du Web sémantique”, Actes des deuxièmes assises du GdR 13, pp 59-78, Cépaduès Editions.
- [2] <http://linkeddata.org/>
- [3] Berners-Lee, J. Hendler, and O. Lassila (2001, May) “The Semantic Web: A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities” *Scientific Am.*, vol. 284, no. 5, pp 28-37.
- [4] Phan Quang Trung Tien Promotion X, (Juil. 2005) “Ontologies et Web Services ”Hanoi.
- [5] <https://fr.slideshare.net/OpenDataSupport/introduction-to-rdf-sparql-v013-fr/>
- [6] <http://www.iro.umontreal.ca/~lapalme/ift6281/EricCharton.pdf>
- [7] Jean Charlet, Philippe Laublet & Chantal Reynaud (Déc. 2003) “le Web sémantique ”Rapport final. Action spécifique 32 CNRS / STIC. J. Charlet, P. La ublet et C. Reynaud. p. 143. 2003
- [8] Baget J.F., Étienne C., Jérôme E. et Mohand S.H. (2004) “Les langages du Web sémantique». *Revue I3, Hors Série 2004*
- [9] "Spécification XML 1.0" . Consortium World Wide Web. Récupéré le 22 août 2010.
- [10] "Extensible Markup Language (XML) 1.0" . *www.w3.org*.
- [11]"Chronologie de normes du Web et du Web sémantique W3C" (PDF). *Dblab.ntua.gr*. Récupéré le 14 août 2016.
- [12] "LICENCE DE DOCUMENT W3C" . *W3.org*. Récupéré le 16 novembre 2017.
- [13] "Origine et objectifs XML 1.0" . *W3.org*. Récupéré le 14 août 2016.
- [14] Fennell, Philip (juin 2013). "*Extrêmes de XML*". XML London 2013 : 80–86. doi : 10.14337 / XMLLondon13.Fennell01 . ISBN 978-0-9926471-0-0.
- [15] <https://wdi.supelec.fr/appliouaibe/Cours/XML>
- [16] <https://www.w3.org/TR/rdf-primer/fig1dec16.png>
- [17] <https://www.w3.org/TR/rdf-primer/>
- [18] Thomas R. Gruber, *Towards Principles for the Design of Ontologies Used for Knowledge Sharing in Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, 1993, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.6025&rep=rep1&type=pdf>

- [19] Tom Gruber, article « Ontology » dans l'*Encyclopedia of Database Systems*, Springer-Verlag, 2009, <http://tomgruber.org/writing/ontology-definition-2007.htm>.
- [20] <http://www.w3.org/TR/2012/REC-owl2-overview-20121211/>
- [21] <https://bu.umc.edu.dz/theses/informatique/HEM6086.pdf>
- [22] <https://fr.slideshare.net/BorderCloud/introduction-au-web-des-donnees-linked-data>
- [23] https://www.researchgate.net/figure/Architecture-of-the-SPARQL-query-mechanism-for-distributed-RDF-repositories-Central_fig1_242319035
- [24] Camille Pradel, Ollivier Haemmerlé, Nathalie Hernandez, Passage de la langue naturelle à une requête SPARQL dans le système SWIP.
- [25] Weiguo Zheng, Lei Zou, Xiang Lian and al, How to Build Templates for RDF Question/Answering—An Uncertain Graph Similarity Join Approach.
- [26] N. Nakashole, G. Weikum, and F. M. Suchanek. Patty: A taxonomy of relational patterns with semantic types. In EMNLP-CoNLL, pages 1135–1145, 2012.
- [27] J Ding et al : The Semantic Web – ISWC 2019: 18th International Semantic Web ..., Partie 1, Page 176, 2019.
- [28](en) Lire par exemple *What's in a link: Foundations for semantic network* , W Woods, BBN Reoirt 3072
- [29] Voir notamment le Workshop de la conférence LREC 2004 Beyond Named Entity Recognition Semantic labelling for NLP tasks
- [30](en) Lire par exemple Wikify! Linking Documents to Encyclopedic Knowledge ,Rada Mihalcea et Andras Csomai, CIKM'07, November 6–8, 2007, Lisboa, Portugal.
- [31](en) Charton E., Gagnon M., Ozell B., Automatic SemanticWeb annotation of named entities, dans Canadian AI (2011)
- [32](en) Mendes P., Jakob M., Garcia-Silva A., Bizer C., DBpedia Spotlight: Shedding Light on the Web of Documents, dans 7th International Conference on Semantic Systems (I-Semantics), vol. Proceeding. Facultad de Informatica (UPM) (2011)
- [33](en) Eric Charton et Juan-Manuel Torres-Moreno, "NLGbAse: a free linguistic resource for Natural Language Processing systems", Proceedings of LREC, 2010.

- [34] Sagot, Benoît, and Rosa Stern. "Aleda, a free large-scale entity database for French." Proceedings of LREC 2012 (2012).
- [35] Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ogniano, D., Goranov, M.: Kim-semantic annotation platform. Lecture Notes in Computer Science p. 834{ 849 (2003)
- [36] (en) Maynard D., Peters W., Li Y., Metrics for evaluation of ontology-based information extraction. In WWW 2006 Workshop on Evaluation of Ontologies for the Web, 2006.
- [37] (en) Lire notamment cette interview d'un chercheur qui détaille le mode de fonctionnement des annotateurs sémantiques dans un contexte multilingue: *Wikimeta Project's Evolution Includes Commercial Ambitions and Focus On Text-Mining, Semantic Annotation Robustness* Sur Semanticweb.com
- [38] MARF Research and Development Group, Serguei Mokhov, Stephen Sinclair, « *Modular Audio Recognition Framework* » MARF, *The Modular Audio Recognition Framework, and its Applications*, sur <http://marf.sourceforge.net/>
- [39] <https://www.yumpu.com/fr/document/read/33356371/corpus-de-dialogue-homme-machine-annotation->
- [40] Valery., P. (2004). "Proposition d'une méthode d'ingénierie ontologique pour les EIAH : application aux systèmes auteurs". Programme de doctorat en informatique, Mai 2004 Université du Québec à Montréal Canada.
- [41] Gruber., T.R. (1995). "Toward principles for the design of ontologies used for knowledge sharing". International Journal of Human Computer Studies.
- [42] Guarino., N. et Giaretta., P. (1995). "Ontologies and knowledge bases: Towards a terminological clarification". In N MARS, réd., Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pp. 25–32. IOS Press.
- [43] Uschold., M. et Grüninger., M. (1996). "Ontologies: principles, methods, and applications". Knowledge Engineering Review, Vol.11, N°2, pp. 93–155.
- [44] Roche., C. (2005). " Terminologie et Ontologie". Revue Langages, N° 157, pp. 48– 62, Editions Larousse.
- [45] Oberle., D. (2006). "Semantic Management of Middleware ". Semantic Web And Beyond Computing for Human Experience: Springer US.
- [46] Paquette., G. (2007). "Graphical Ontology Modeling Language for Learning Environments". Technology, Instruction., Cognition and Learning, Vol. 5, pp. 133– 168.
- [47] Héon., M. (2010). "OntoCASE: Méthodologie et assistant logiciel pour une ingénierie ontologique fondée sur la transformation d'un modèle semi-formel". Thèse de doctorat, Université du Québec, Montréal

Résumé

Ce travail est présenté dans le cadre d'une mémoire en vue d'obtenir le diplôme de Master 2 Académique, dont le but est de concevoir et réaliser un système d'apprentissage intitulé « Vers un système d'apprentissage basé sur linked data » à pour but du traitement de données est généralement d'offrir une information de plus haut niveau ou une information de meilleure qualité à un autre outil de traitement ou d'analyse traiter la connaissance elle-même, au lieu du texte, en utilisant des procédés similaires à un raisonnement déductif humain et de l'inférence, obtenir des résultats plus significatifs tout en aidant les ordinateurs à effectuer de la collecte d'informations automatisée et ainsi faciliter et améliorer la recherche.

Ainsi l'ère des plates-formes collaboratives construites dans des domaines différents (gestion des connaissances, enseignement à distance, Télémédecine, etc.), intègrent plusieurs outils de collaboration (synchrones et asynchrones) et offrent des possibilités d'interaction et d'échange d'informations entre des groupes de personnes situés dans des endroits différents.

Mots clés : Site web, Web Sémantique, données liées

Abstract

This work is presented as part of a thesis in order to obtain the diploma of Master 2 Academic, the aim of which is to design and implement a learning system entitled "Towards a learning system based on linked data", with the aim of treating data is usually to offer higher level information or information higher quality to another processing or analysis tool process the knowledge itself, instead of the text, using processes similar to deductive reasoning human rights and inference, achieve more meaningful results while helping computers to perform automated information collection and thus facilitate and improve research. Thus the era of collaborative platforms built in different areas (knowledge management, distance learning, Telemedicine, etc.), integrate several collaboration tools (synchronous and asynchronous) and offer possibilities interaction and exchange of information between groups of people located in different places.

Keywords: Website, Semantic Web, linked data