

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de 8 Mai 1945 – Guelma -
Faculté des Mathématiques, d'Informatique et des Sciences de la matière
Département d'Informatique



Mémoire

Filière : Informatique

Option : STIC

Thème :

Vers la transformation de l'université en un campus intelligent. Etude de cas : Université de Guelma

Encadré Par :
Djakhdjakha Lynda

Présenté par :
Cisse Hamadoun

Juin 2022

Remerciements

Louange à Allah, de me donner la force et la patience de survivre, ainsi que le courage de surmonter toutes les difficultés pour atteindre la fin de mon master et compléter ce modeste travail.

Ainsi j'adresse mes plus sincères remerciements et ma profonde reconnaissance à **Mme. Djakhdjakha Lynda** mon encadreuse de mémoire de Master pour la pertinence de ses remarques et sa patience pendant ce travail, sa manière de diriger qui fut pour moi une grande source d'inspiration et de motivation, elle a su m'encourager et me permettre de travailler dans une ambiance scientifique exceptionnelle.

J'exprime toute ma reconnaissance à Monsieur **Mr. Hallaci Samir** pour son aide et ses conseils précieux.

Je tiens à remercier **Mr. Farou Ibrahim** pour tous ses efforts pour l'aboutissement de ce travail.

Mes remerciements vont aussi à **Mr. Zineddine Kouahla**, à **Mme. Benhamza Karima**, à **Mme Mehnoui Zahra**, à **Mme. Nadjette Benhamida**.

Enfin, Je remercie tous mes enseignants du début à la fin de mon cycle de Master,

Merci pour toute l'équipe pédagogique du département d'informatique de l'université 8 Mai 1945 de Guelma, et je remercie toutes les personnes qui m'ont aidée, de près ou de loin pour la réalisation de ce travail en particulier.

Dédicaces

Je dédie ce mémoire à :

Ma mère, qui a œuvré pour ma réussite, par son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance, reçois à travers ce travail modeste, l'expression de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

Mr. Soumeylou Boubèye Maïga, grâce à qui j'ai eu l'opportunité de faire mon master en Algérie. Que son âme repose en paix.

Mes sœurs et mes frères, votre soutien et encouragement me marqueront à jamais

Mes amis, Je mentionne particulièrement **Abdoul Aziz Diallo** qui n'ont cessé d'être pour moi une source de motivation inépuisable.

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible,
je vous dis merci.

Résumé

Dans les dernières années, la majorité des universités intelligentes ont pris note « qu'un environnement innovant favorise une culture innovante ». Elles ont remarqué les avantages de la mise en œuvre de l'IoT pour les économies d'énergie, l'amélioration de la sécurité, etc. Les experts estiment que la taille du marché mondial de l'IoT dans le domaine de l'éducation devrait atteindre 11,3 milliards de dollars d'ici 2023 grâce (i) à l'utilisation accrue des objets connectés dans les institutions universitaires, (ii) à l'adoption rapide d'outils d'apprentissage en ligne et (iii) à la grande disponibilité de solutions basées sur les nouvelles technologies de l'information et le cloud computing.

C'est dans ce cadre et dans le but de créer « un smart campus » que se situe ce travail qui vise (i) à utiliser l'Internet des objets, la fusion et l'exploration de données, le Cloud Computing, les technologies du web sémantique et d'autres technologies de l'information et (ii) à mettre l'accent sur l'intégration des systèmes d'informations indépendants, des données hétérogènes et de leurs analyses dans une volonté d'apporter des meilleurs services à la communauté universitaire.

Mots-clés :

Système IoT, Smart Campus, Université intelligente, Campus intelligent, Ontologie, Ontologie modulaire, IoT, Hadoop, Hive, Big Data.

Abstract

In recent years, smart universities have realized that ‘*an innovative environment generates an innovative culture*’. Universities have recognized the advantages of integrating IoT for energy efficiency, improved security, and other objectives. Experts estimate that the global IoT in education market size will expand 11,3 billion dollars by 2023, due mainly to (i) the increased use of connected objects in academic institutions, (ii) rapid adoption of e-learning tools and (iii) the wide availability of solutions based on new information technologies and cloud computing.

This work is positioned within this framework and with the vision of making a “smart campus”, with the goals of (i) utilizing the IoT, data fusion and exploration, cloud computing, semantic web technologies, and other information technologies, and (ii) highlighting the integration of independent information systems, heterogeneous data, and their analyses in order to satisfy the university community.

Key words. IoT system, Smart Campus, Smart University, Ontology, Modular Ontology,

المخلص

أدركت الجامعات الذكية في السنوات الأخيرة أن "البيئة المبتكرة تولد ثقافة مبتكرة". ولقد أدركت الجامعات مزايا دمج إنترنت الأشياء من أجل كفاءة الطاقة وتحسين الأمن والأهداف الأخرى.

ويقدر الخبراء أن حجم إنترنت الأشياء العالمي في سوق التعليم سيتوسع بمقدار 11.3 مليار دولار بحلول عام 2023، ويرجع ذلك أساساً إلى (1) زيادة استخدام الكائنات المتصلة في المؤسسات الأكاديمية، (2) الاعتماد السريع لأدوات التعلم الإلكتروني و (3) التوفر الواسع للحلول القائمة على تقنيات المعلومات الجديدة والحوسبة السحابية.

في هذا الإطار، يرمي هذا المشروع إلى إنشاء "حرم جامعي ذكي"، بهدف (1) استخدام إنترنت الأشياء، ودمج البيانات واستكشافها، والحوسبة السحابية، وتقنيات الويب الدلالية، وتقنيات المعلومات الأخرى، و (2) إبراز تكامل نظم المعلومات المستقلة والبيانات غير المتجانسة وتحليلاتها بما يرضي مجتمع الجامعة.

الكلمات الدالة. نظام إنترنت الأشياء، الحرم الجامعي الذكي، الجامعة الذكية، الأنطولوجيا، الأنطولوجيا المعيارية،

Sommaire

Remerciements	I
Dédicaces	II
Résumé	III
Abstract	IV
الملخص	V
Sommaire	VI
Liste des Tableaux.....	IX
Liste des Algorithmes.....	X
Liste des Figures	XI
Liste des Abréviations	XIII
Introduction Générale.....	1
Chapitre 1 : Etat de l'art	4
Introduction :	4
1 Concepts de base :	4
1.1 Campus Intelligent :	4
1.2 Ontologie :	5
1.3 Architecture Cloud :	7
1.4 Architecture Fog :	8
1.5 Architecture Edge :	8
1.6 Algorithme Génétique :	9
Conclusion :	11

Chapitre 2 Etude des Travaux connexes.....	12
Introduction :.....	12
1 Travaux Connexes :.....	12
Conclusion :.....	23
Chapitre 3 : Conception.....	24
Introduction :.....	24
1 Première contribution : La génération de l’emploi du temps :.....	24
1.1 Les contraintes pour la génération de l’emploi du temps :.....	24
1.2 Structure de donnée :.....	25
1.3 Le codage :.....	26
1.4 Population initiale :.....	26
1.5 La fonction objective :.....	27
1.6 L’algorithme de sélection :.....	27
1.7 L’opération de croisement :.....	27
1.8 L’opération de mutation :.....	28
1.9 Critères d’arrêt :.....	28
2 Deuxième contribution : La gestion d’un campus intelligent basée sur une ontologie modulaire et architecture Cloud-Fog-Edge :.....	29
2.1 Conception de l’Ontologie :.....	29
2.1.1 Les modules de l’ontologie campus intelligent :.....	29
2.1.2 Structuration et développement des modules du campus intelligent :.....	31
2.1.2.1 Structuration :.....	31
2.1.2.2 Développement des modules :.....	31
2.1.2.2.1 Le module « Acteur » :.....	32
2.1.2.2.2 Le module « Classe intelligente » :.....	36
2.2 Architecture du Système :.....	39

Conclusion :	41
Chapitre 4 : Implémentation	42
Introduction :	42
1 Outils :	42
1.1 MQTT :	42
1.2 Flask :	42
1.3 Owready2 :	42
2 Implémentation de la génération de l'emploi du temps :	42
3 Implémentation de l'ontologie :	51
3.1 Module « Acteur » :	51
3.2 Le module « Classe intelligente » :	53
4 Architecture du système :	56
Conclusion :	58
Conclusion Générale	59
Références	60

Liste des Tableaux

Tableau 2.1: Synthèse des travaux connexes	22
Tableau 3.1 : Représentation du chromosome	26
Tableau 3.2 : Liste des concepts, relations, attributs Module « Acteur »	36
Tableau 3.3 : Listes des concepts, relations et attributs du Module « Classe Intelligente »	39

Liste des Algorithmes

Algorithme 1 : Initialisation de la population	227
--	-----

Liste des Figures

Figure 1.1 : Campus intelligent	4
Figure 1.2 : Exemple d'ontologie pour un parking [26]	6
Figure 1.3 : Architecture Cloud [30]	7
Figure 1.4 : Architecture Fog [30]	8
Figure 1.5 : Architecture Edge [30]	9
Figure 1.6 : Principe de fonctionnement de l'algorithme génétique [36]	10
Figure 3.1 : Illustration de l'opération de croisement	28
Figure 3.2 : Illustration de l'opération de mutation	28
Figure 3.3 : Structure des modules	31
Figure 3.4 : Diagramme de classe Module « Acteur »	33
Figure 3.5 : Diagramme de classe du module « Classe Intelligente »	37
Figure 3.6 : Architecture proposée	40
Figure 4.1 : Extrait du code de l'algorithme génétique	43
Figure 4.2 : Page d'accueil de l'application	44
Figure 4.3 : Connexion d'un enseignant	44
Figure 4.4 : Page d'accueil de l'enseignant	45
Figure 4.5 : Menu des vœux de l'enseignant	45
Figure 4.6 : Enregistrement d'un vœu	46
Figure 4.7 : Affichage du vœu enregistré	46
Figure 4.8 : Remplissage complet Fiche de vœux	47

Figure 4.9 : Connexion du chef de la pédagogie.....	47
Figure 4.10 : Page d'accueil du chef de la pédagogie.....	48
Figure 4.11 : Première étape création emploi du temps	48
Figure 4.12 : Menu création emploi du temps	49
Figure 4.13 : Erreur en cas de duplication	49
Figure 4.14 : Affichage de l'emploi du temps générée.....	50
Figure 4.15 : Impression de l'emploi du temps	50
Figure 4.16 : Code de création de l'ontologie Acteur.....	51
Figure 4.17 : Détails de l'ontologie Acteur	51
Figure 4.18 : Hiérarchie des classes de l'ontologie Acteur	52
Figure 4.19 : Hiérarchie des relations de l'ontologie Acteur.....	52
Figure 4.20 : Hiérarchie des attributs de l'ontologie Acteur	53
Figure 4.21 : Code de création de l'ontologie Classe intelligente	53
Figure 4.22 : Détails de l'ontologie Classe intelligente	54
Figure 4.23 : Hiérarchie de classe de l'ontologie Classe intelligente.....	54
Figure 4.24 : Hiérarchie de relation de l'ontologie Classe Intelligente	55
Figure 4.25 : Hiérarchie des attributs de l'ontologie Classe intelligente.....	55
Figure 4.26 : Lecteur intelligent de carte développé par nous	56
Figure 4.27 : Capteur de température.....	56
Figure 4.28 : Interrupteur de fin de course.....	57
Figure 4.29 : Capteur ultrason.....	57
Figure 4.30 : Routeur	58

Liste des Abréviations

AIIISO	A cademic I nstitution I nternal S tructure O ntology
API	A pplication P rogramming I nterface
FOAF	F riend of a friend
HTTP	H ypertext T ransfer P rotocol
IaaS	I nfrastucture as a S ervice
IoT-O	I nternet of T hings ontology
KM-KNN	K Means – K Nearest Neighbor
MQTT	M essage Q ueuing T elemetry T ransport
OAM	O ntology A pplication M anagement
OEMA	O ntology for E nergy M anagement A pplications
ONGSC	O ntology of N ext G eneration S mart C lassrooms
OWL	O ntology W eb L anguage
OWL-DL	O ntology W eb L anguage D escription L ogics
PaaS	P latform as a S ervice
QB	Q uery B uild
RDF	R esource D escription F ramework
RFID	R adio F requency I dentification
SaaS	S oftware as a S ervice
SPARQL	S imple P rotocol and R DF Q uery L anguage
SQWRL	S emantic W eb Q uery L anguage
SSN	S emantic S ensor N etwork
SWRL	S emantic W eb R ule L anguage
Wise-IoT	W ise I nternet of T hings
XML	e Xtensible M arkup L anguage

Introduction Générale

Contexte Générale et Problématique

Lors de ces dernières années, l'évolution des nouvelles technologies a entraîné la naissance d'un nouveau concept dit « Campus intelligent ». Un campus intelligent est une tendance émergente qui permet aux établissements d'enseignement de combiner les technologies intelligentes avec leur infrastructure physique pour améliorer les services, la prise de décision, la durabilité du campus, etc [1]. Elle permet de mettre en œuvre des solutions au niveau des campus, telles que des micro-réseaux intelligents, des salles de classe intelligentes, le contrôle des propriétés visuelles et thermiques des bâtiments, le contrôle de la présence des étudiants par reconnaissance faciale/carte intelligente, etc [1]. Son intelligence se base sur un ensemble de capteurs interconnectés. Par exemple, au niveau des salles de cours, des capteurs permettent de réduire la consommation d'électricité par l'arrêt des climatiseurs et des lumières lorsque les salles et les classes sont vides. Cependant, malgré ces avantages irréfutables, le développement d'un campus intelligent est une entreprise très complexe et difficile [2]. Les données générées par le nombre colossal de capteurs utilisées dans un campus intelligent présentent différents problèmes. Ses données issues de sources (capteurs) différentes, sont de natures différentes (structurées, non-structurées), de grande taille et de sémantique faible. De plus, nos outils traditionnels de gestion de données sont obsolètes face à ce phénomène. Ainsi la mise en place d'un système efficace et optimal pour le stockage, l'intégration, le traitement et l'exploitation de ses données est un véritable défi.

Pour faire face à ces défis, notre objectif est de proposer un système qui inclue l'ensemble des modules nécessaires pour transformer l'université en un campus intelligent.

Contributions

De ce fait, nous avons commencé par l'amélioration de notre travail de licence 'Un système de gestion de la pédagogie université 08 mai Guelma'. L'amélioration consiste à proposer une génération automatique de l'emploi du temps qui constitue le module de base d'une gestion manuelle ou intelligente de la pédagogie. Pour la génération nous avons utilisé l'algorithme génétique car elle s'avère être la méthode la plus utilisée pour les problèmes de génération de l'emploi du temps [3]–[6].

Ainsi, nous avons décidé de modéliser nos données sous forme d'ontologie vu leurs expressivités et large acceptation dans la communauté scientifique, outre les possibilités d'inférences et de raisonnements qu'offre une telle modélisation [7]. De plus, elle permet l'automatisation de différentes activités et processus dans l'enseignement supérieurs en utilisant le raisonnement automatique [8]. De plus elle permet l'intégration de nos données tout en les ajoutant une sémantique. Elle permet aussi d'améliorer la précision de la recherche d'information, en évitant des ambiguïtés au niveau terminologique (Recherche sémantique) [9]. Pour la conception de notre ontologie nous avons opté pour une ontologie modulaire car elle permet réduire la taille des composants de l'ontologie, faciliter la maintenance et la validation de l'ontologie, faciliter une meilleure compréhension, l'évolution, et le coût de création de l'ontologie [10]–[12].

Nous avons décidé d'utiliser une architecture Cloud, Fog, Edge pour la mise en place de notre système. Ce choix s'est effectué après avoir envisagé plusieurs architectures possibles en vue de concevoir un système capable de stocker et de traiter l'ensemble des données massives issues du campus intelligent dans un temps réel tout en évitant les surcharges sur les réseaux. Parmi les différentes architectures on distingue une architecture Cloud, une architecture Fog et une architecture Edge. Les trois peuvent être utilisés séparément ou combinés ensemble pour produire une architecture de système. Utiliser une architecture Cloud permet de répondre à notre besoin de gérer un grand volume de données [13]. Avec notre système nous travaillons avec des données issues de capteurs et à partir de ses données nous devons effectuer des traitements en temps réel (actionner l'ouverture de la porte lors de l'identification de l'utilisateur). Cependant avec l'architecture Cloud, l'envoi des données collectées au serveur à chaque fois, les traiter et ensuite envoyer la commande d'ouverture serait très coûteux en termes de bande passante et en temps de réponse [14]. De plus en cas d'absence d'internet les données ne seront pas accessibles au niveau du Cloud. Pour remédier à cela, l'architecture Edge serait la solution pour le besoin en temps réel [15]. Il permet de rapprocher les traitements près des sources de données ce qui permet d'accélérer considérablement le temps de traitement. L'inconvénient de ce type d'architecture est que la capacité de stockage et de traitement y est limitée. Elle est surchargée très facilement à cause des grandes quantités de données [16]. Le choix probable serait d'utiliser une architecture Fog qui est l'intermédiaire entre le Cloud et le Edge. Elle offre un temps de réponse moins long que celui du Cloud [17]. Le Cloud reste plus puissant que le Fog en ce qui concerne les capacités de calcul et de stockage [18].

Plan de Travail

Ce mémoire sera structuré en quatre chapitres.

Dans le premier chapitre on fait une présentation des concepts clés de notre projet et dans le second chapitre on s'intéresse à la synthèse des différents travaux connexes.

Le troisième chapitre porte sur la conception de nos différentes contributions.

Le dernier chapitre porte l'implémentation du système.

Chapitre 1 : Etat de l'art

Introduction :

Dans ce chapitre, nous allons évoquer les concepts clés de notre mémoire qui sont le campus intelligent, l'ontologie, les différentes architectures et l'algorithme génétique. Nous essayerons de donner une description brève de ses concepts.

1 Concepts de base :

1.1 Campus Intelligent :

D'après [19], un campus intelligent est constitué d'un ensemble de six (6) sous domaines : classe intelligente, parking intelligent, apprentissage intelligent, gestion intelligente, santé intelligente et gouvernance intelligente.

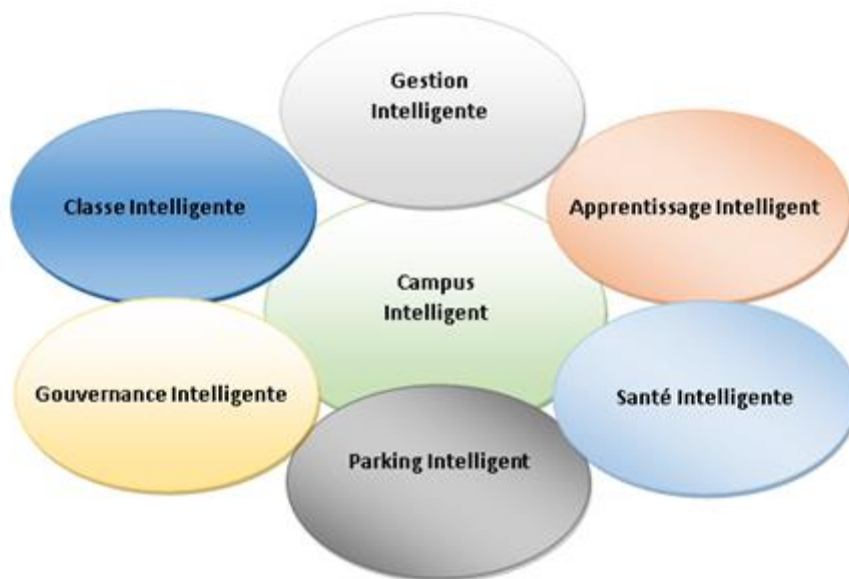


Figure 1.1 : Campus intelligent

- Une classe intelligente est essentiellement une salle de classe améliorée par la technologie qui utilise l'informatique pour faciliter l'apprentissage chez les élèves [20]. Il vise à améliorer les interactions entre les élèves et les enseignants en termes de niveau et de qualité [20]. Aussi, il fournit des informations concernant différentes conditions dans la classe, telles que la température de la pièce, la lumière allumée ou éteinte dans la pièce et le nombre d'étudiants présents [19].

- L'apprentissage intelligent est un sous-domaine qui est lié à différents aspects de l'apprentissage dans un campus et comprend l'apprentissage collaboratif, l'apprentissage en ligne, l'apprentissage personnalisé, et les classes virtuelles [19]. Il soutient les étudiants et le corps enseignant dans leur tâche d'acquisition de connaissances [21]. Cela inclut la fourniture de moyens dans la préparation et la transmission des contenus, mais aussi (et peut-être plus important encore) de fournir des moyens aux étudiants d'apprendre, individuellement ou en collaboration, et d'accéder aux contenus pertinents en tout lieu et à tout moment [21].
- La gestion intelligente est un autre sous-domaine qui concerne les capacités de gestion du campus, facultés, départements et personnes [19]. Elle englobe les aspects plus physiques d'un campus, comme la gestion intelligente des bâtiments, l'accès et le contrôle des étudiants, la sécurité et la surveillance, ainsi que la réponse aux urgences [21].
- La santé intelligente fournit des soins de santé préventifs, des soins et un suivi à distance et des systèmes d'alerte aux épidémies [21].
- La gouvernance intelligente est un sous-domaine qui intègre les politiques régissant un campus intelligent afin d'assurer l'efficacité et l'efficience continues dans l'environnement global du campus [19].
- Le parking intelligent fournit des informations sur les places de stationnement disponibles, et qui fournit également des informations lorsque le parking est plein [22].

1.2 Ontologie :

Le concept d'ontologie est utilisé dans beaucoup de domaine. Ainsi il existe de multiples définitions au concept Ontologie [23]. Une ontologie est définie comme "une structure de données pour conceptualiser des connaissances" [19]. Dans [24], on le considère comme des théories de domaine qui spécifient un vocabulaire spécifique d'entités, de classes, de propriétés, de prédicats et de fonctions, et comme un ensemble de relations qui existent nécessairement entre ces termes de vocabulaire. Pour d'autres c'est une description formelle des connaissances, c'est un ensemble de concepts dans un domaine et les relations entre eux [25]. Le terme "formel" signifie dans un langage dont la syntaxe et la sémantique sont bien définies. La figure ci-dessous décrit un exemple d'ontologie. Il s'agit de la représentation d'un parking. Les blocs

en violet représentent les concepts, les blocs en vert représentent les instances et les flèches représentent les relations entre les concepts.

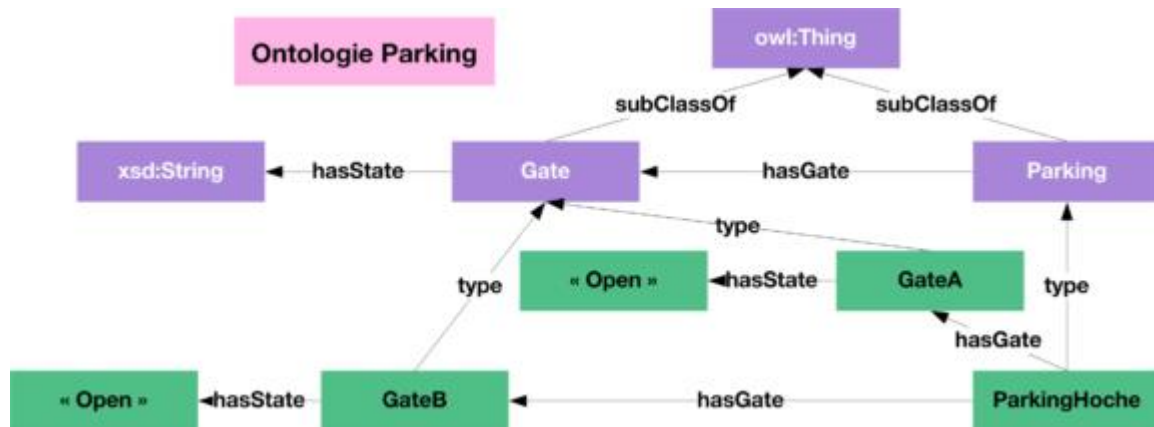


Figure 1.2 : Exemple d'ontologie pour un parking [26]

Une ontologie est composée de : concepts, propriétés, relations, axiomes et des instances [27]. Elle est largement utilisée en ingénierie des connaissances et l'intelligence artificielle pour structurer les concepts d'un domaine [19]. On utilise les ontologies pour différentes raisons. L'une de ses principales raisons est la possibilité de raisonner automatiquement sur les données. On les utilise aussi pour d'autres raisons [25], [27] :

- Acquérir et représenter les connaissances.
- Rechercher et faire l'extraction des connaissances.
- Partager une compréhension commune de la structure de l'information entre des personnes ou des agents logiciels.
- Réutiliser la connaissance du domaine en représentant les concepts et leurs relations.
- Représenter tous les formats de données, y compris les données non structurées, semi-structurées ou structurées, ce qui facilite l'intégration des données, l'exploration conceptuelle et textuelle et l'analyse des données.

Les ontologies bénéficient de plusieurs langages de représentation (tels que OWL, XML et RDF) et d'outils d'inférences, à travers les règles SWRL (Semantic Web Rule Engine), SQWRL (Semantic Web Query Language), et SPARQL (SPARQL Protocole and RDF Query Language) [28].

1.3 Architecture Cloud :

Le cloud computing ou informatique en nuage est une infrastructure dans laquelle la puissance de calcul et le stockage sont gérés par des serveurs distants auxquels les usagers se connectent via une liaison Internet sécurisée [29].

Ce type d'architecture est très efficace pour effectuer des calculs sur de grand volume de données.

On distingue plusieurs types de services offerts par le Cloud [29]:

- IaaS (Infrastructure as a Service) : le système d'exploitation et les applications sont installés par les clients sur des serveurs auxquels ils se connectent pour travailler comme s'il s'agissait d'un ordinateur classique.
- PaaS (Platform as a Service) : dans ce mode, c'est le fournisseur du service cloud qui administre le système d'exploitation et ses outils. Le client peut installer ses propres applications si besoin.
- SaaS (Software as a Service) : les applications sont fournies sous forme de services clés en mains auxquels les utilisateurs se connectent en utilisant des logiciels dédiés ou un navigateur Internet.

Le temps de latence pour l'architecture Cloud est élevé et imprévisible, et il entraîne des surcharges de réseau [30].

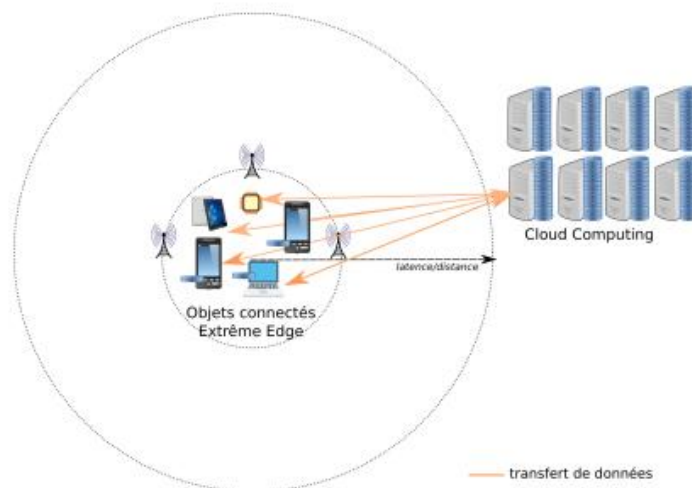


Figure 1.3 : Architecture Cloud [30]

1.4 Architecture Fog :

Le Fog Computing peut être défini comme l'ensemble de dispositifs informatiques de traitement et de stockage de données, servant de rapprocher le Cloud Computing des appareils qui produisent et agissent sur les données [31].

C'est une architecture intermédiaire en l'architecture Cloud et l'architecture Edge.

Il a un temps de latence beaucoup moins courts que celui du Cloud [30].

C'est une architecture décentralisée par rapport au Cloud qui est une architecture centralisée.

Cependant le Cloud a un meilleure capacité de stockage et calcul [32].

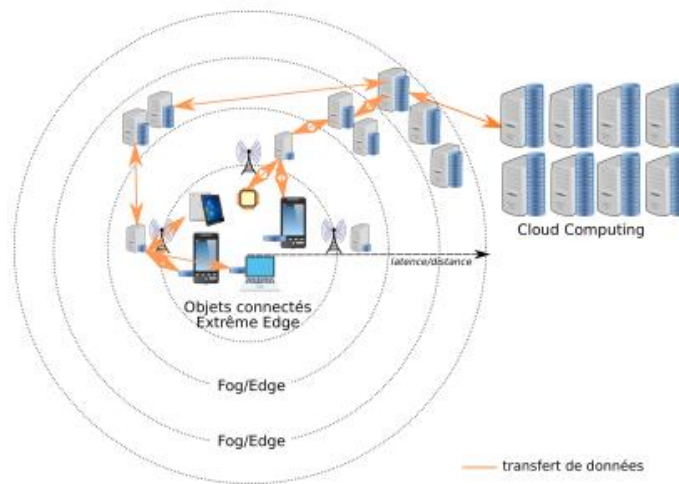


Figure 1.4 : Architecture Fog [30]

1.5 Architecture Edge :

L'Edge computing ou « informatique en périphérie du réseau » désigne le traitement informatique réalisé à l'emplacement le plus proche des données d'un système ou de son utilisateur final, là où les informations sont émises et reçues [33].

C'est une architecture qui permet de réduire le temps de latence du traitement des données.

Les ressources des unités calculs et de stockage sont limités au niveau de l'architecture Edge [30].

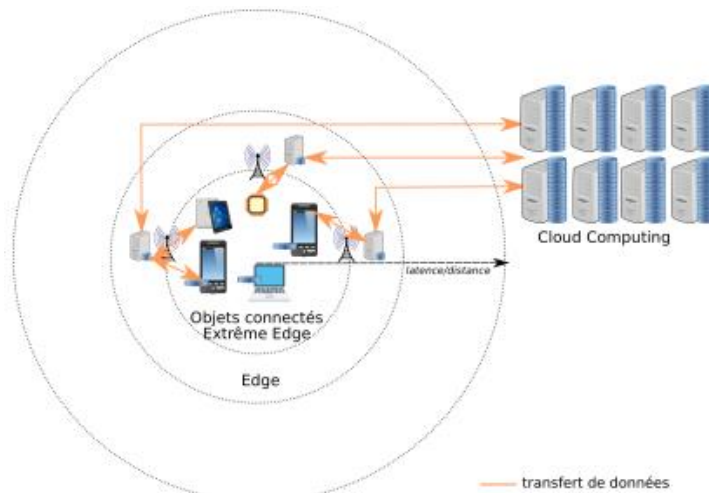


Figure 1.5 : Architecture Edge [30]

1.6 Algorithme Génétique :

L'algorithme génétique est une technique d'optimisation basée sur les concepts de la sélection naturelle et génétique [34]. Ils sont efficacement utilisés pour résoudre des problèmes à contraintes multiples.

Le but principal des algorithmes génétiques est de trouver une solution pour des problèmes difficiles c'est-à-dire des problèmes où on ne connaît pas de méthodes exactes pour les résoudre en temps raisonnable [35]. La solution obtenue peut être la solution optimale ou une solution approchée de l'optimale.

Avant d'aborder le principe de fonctionnement de l'algorithme, il est important de définir quelques terminologies utilisées dans les algorithmes génétiques :

Chromosome : Il correspond à un élément représentant une solution possible à un problème donné.

Gène : Chaque chromosome est divisé en un ensemble d'unités appelées gènes.

Individu : Il s'agit d'un chromosome.

Population : Elle est définie comme l'ensemble des chromosomes. On l'appelle aussi une génération.

Parents : Ils correspondent aux individus pouvant donner naissance à de nouveaux individus afin de former une nouvelle génération.

Le principe de fonctionnement de l'algorithme génétique est décrit par la figure ci-dessous.

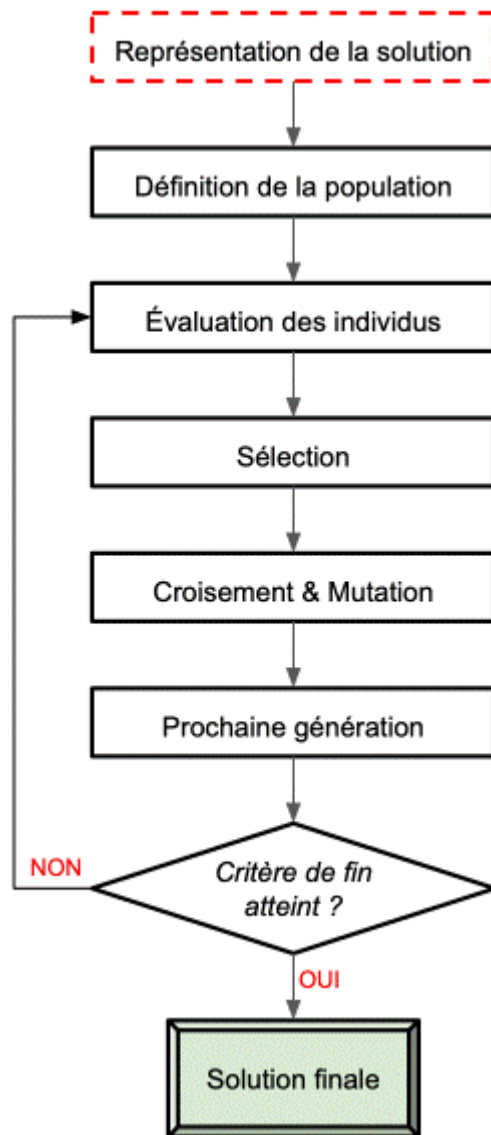


Figure 1.6 : Principe de fonctionnement de l’algorithme génétique [36]

Le fonctionnement de l’algorithme génétique comprend les étapes suivantes :

- **Représentation de la solution (codage) :**

Elle permet de modéliser les données du problème réel en données utilisables par l’algorithme génétique. La qualité du codage des données conditionne le succès de l’algorithme génétique [37].

- **Définition de la population :**

Il s’agit de la phase de génération de la population initiale. Il s’agit du point de départ de l’algorithme et son choix influe sur la rapidité et la qualité de la solution finale [35].

- **Evaluation des individus :**

L'évaluation permet de s'assurer que les individus performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population [34]. Pour cela on utilise une fonction objective qui permet d'évaluer une solution afin de la comparer aux autres.

- **La sélection :**

Elle permet de garantir de choisir les chromosomes qui permettront d'améliorer la qualité des solutions. Il existe plusieurs méthodes de sélection : la sélection par roulette, la sélection par rang ou encore la sélection par tournoi.

- **Les opérateurs de croisement et de mutation :**

Ils permettent de faire évoluer une population tout en améliorant la fonction objective. Cela permet de générer de nouvelles solutions. L'opérateur de croisement recompose les gènes d'individus existant dans la population, alors que l'opérateur de mutation a pour but de garantir l'exploration de l'espace de recherche en introduisant des nouveaux individus [37].

- **Les paramètres de dimensionnement :**

Il s'agit de la taille de la population, du nombre total de générations (critères d'arrêt), probabilités d'application des opérateurs de croisement et de mutation, etc.

Conclusion :

Nous avons présenté les concepts de bases que nous avons utilisé dans notre travail. La présentation de ces différents concepts a été brève sans être trop détaillée mais permettant d'avoir un aperçu général. Dans le chapitre suivant nous allons effectuer une étude des différents travaux connexes.

Chapitre 2 Etude des Travaux connexes

Introduction :

Dans ce chapitre, nous présenterons les différents travaux relatifs à notre thématique car c'est une étape importante pour la mise en évidence de notre contribution. Nous sommes concentrés d'avantages sur les travaux relatives au campus intelligent intégrant l'usage des ontologies.

1 Travaux Connexes :

Dans la littérature, on retrouve beaucoup de travaux dans le domaine du campus intelligent. Cependant dans cette partie nous allons citer quelques travaux connexes en nous limitant aux travaux récents et qu'utilisent les ontologies.

- [38] propose un système de gestion des affectations des cours au sein de l'université en utilisant les ontologies. Dans leur travail, une ontologie éducative est développée pour modéliser de façon sémantique les cours et les profils académiques dans les universités. Ensuite le système permet d'affecter au cours l'enseignant le plus qualifié en se basant sur son profil académique. Malgré que leur système donne des résultats précis, il y'a un grand nombre important de cours qui ne sont pas affectés à des enseignants.
- [2] a travaillé sur une ontologie pour un système de gestion de campus intelligent pour des enfants. Il utilise une architecture basée sur l'internet des objets. Avec un réseau de capteur permettant le suivie, le système permet de surveiller l'enfant et ses activités en vue déterminer ses talents. Ainsi le corps éducatif pourra fournir une formation plus adaptée à l'enfant. Cependant il n'existe pas une implémentation du système malgré une présentation théorique (graphique et logique).
- [39] se concentre sur un système orientée service de gestion autonome de l'architecture IoT du campus intelligent au moyen d'ontologie. L'objectif est de mettre en œuvre un système qui peut faire face aux changements et aux évolutions tout au long de l'exécution en fonction des spécificités et des besoins dynamiques des utilisateurs du système. L'architecture proposée est innovante car elle s'appuie sur des ontologies pour la détection du contexte. L'implémentation du travail a été déployée au sein de deux campus. L'ontologie proposée réutilise l'ontologie IoT-O.

- Le travail de [40] porte sur une ontologie pour un parking intelligent compatible avec l'IoT dans un campus intelligent. Il utilise les ontologies pour fournir une sémantique en termes de vocabulaire commun et pour faciliter l'échange d'informations, ce qui permet de résoudre les problèmes d'interopérabilité au sein des systèmes intelligents. L'ontologie réutilise des ontologies existantes : Wise-IoT, SSN, FOAF. La méthodologie adoptée pour développer l'ontologie suit une approche agile et modulaire. Cependant leur système n'est que théorique.
- [41] a développé une ontologie pour la gestion d'une salle de classe intelligente utilisant l'IoT dans un campus intelligent. Leur système permet de gérer les problèmes d'interopérabilité sémantique dans un environnement de campus intelligent, en tenant compte de nouveaux paradigmes comme l'IoT et de concepts pédagogiques comme l'analyse de l'apprenant. Par rapport aux ontologies existantes, l'ontologie proposée permet de suivre les progrès de l'étudiant. Plutôt que de concevoir l'ontologie à partir de zéro, l'ontologie réutilise les concepts des ontologies existantes et s'appuie sur des concepts liés aux performances des étudiants et des instructeurs. Ces concepts permettent de suivre les progrès de l'apprenant et de l'instructeur afin d'améliorer le processus d'apprentissage dans une classe intelligente. Il n'y a pas d'implémentation réelle du système. Les ontologies réutilisées sont : Context Ontology, ONGSC, SSN, FOAF.
- [42] propose une ontologie pour la description des cours de l'université. Le travail présente de façon claire le rôle des concepts généralement utilisées dans le domaine des cours. L'ontologie réutilise des ontologies existantes : AIISO, OWL Time, Dublin Core, FOAF. L'implémentation du système permet d'effectuer des tâches telles que trouver l'itinéraire pour une salle de classe ou encore trouver les prérequis pour un cours.
- Le papier de [43] porte sur un système multi-agent utilisant les ontologies pour améliorer la gestion au sein d'une école intelligente. Ainsi le système permet aux administrateurs d'identifier toutes les activités liées au cours et plus précisément la gestion de l'attendance des étudiants et des enseignants, etc. Pour la détection de l'attendance, les auteurs utilisent des dispositifs wifi pouvant communiquer avec les mobiles des utilisateurs. Ainsi la présence de l'étudiant est validée si son mobile se connecte au dispositif wifi installé dans la salle de cours. La précision du système

présente de nombreux problèmes comme le cas de mobile ayant le wifi désactivé ou encore le cas des interférences.

- L'ontologie de [44] permet la représentation des curriculums et des syllabus dans l'enseignement supérieur. Le travail se concentre sur la conception détaillée des différents concepts. De plus les auteurs présentent une méthode d'intégration et de classification du syllabus.
- [45] présente une ontologie pour la représentation du profil de l'étudiant dans un système d'apprentissage gamifié adaptatif. Les auteurs combinent deux concepts 'Apprentissage Adaptatif' et 'gamification' pour fournir une expérience de gamification personnalisée. Leur modèle utilise les caractéristiques académiques, psychologiques et de gamification de l'étudiant. Le système permet de faciliter les prises de décisions dans les différents contextes académiques et de motiver l'étudiant pour achever ses objectifs d'apprentissage.
- [46] propose un système basé ontologie pour conseiller les étudiants sur le choix du programme éducatif. Ce dernier permet à l'étudiant de bien choisir son parcours éducatif. Il prend en compte de nombreuses contraintes telles que la zone géographique, le mérite et le coût afin de bien orienter l'étudiant.
- [47] se concentre sur une ontologie pour l'apprentissage en ligne, plus précisément le domaine de la physique. Les auteurs utilisent une approche hybride qui combine le traitement du langage naturel et le filtrage par motif pour créer une ontologie de façon automatique basée sur des textes obtenus à partir de la presse ou d'Internet comme document du domaine. Cependant tous les textes utilisés pour l'ontologie sont écrits en chinois.
- [48] contribue avec un Framework basé ontologie pour réduire la consommation d'électricité au sein de la librairie d'un campus. Le système utilise des données issues du planning et d'un ensemble de capteurs. Ainsi les données issues des capteurs sont traitées avec une ontologie afin d'inférer les activités des étudiants. A partir de l'activité, le système peut prédire de fermer une zone à un moment donné ou encore d'éteindre les dispositifs électriques afin de réduire la consommation électrique de la librairie.

- Le travail de [49] présente un système de recommandation basé ontologie pour l'enseignement supérieur. En plus de l'ontologie, le système utilise l'apprentissage automatique pour orienter les étudiants. Ainsi il permet d'identifier les besoins des étudiants, leurs intérêts, préférences et capacités afin de recommander la voie ou l'université la plus approprié.
- Un système basé sur les ontologies a été proposé par [25] pour la coordination des soins dans une école intelligente pour les enfants et les jeunes ayant des besoins spéciaux ou des handicaps. Parmi les besoins, on retrouve par exemple le cas des étudiants autistes qui ont besoin d'informations concrètes et visuelles (objets, photos, ...) pour comprendre. Le système permet la coordination, l'intégration et le suivi des activités (enseignement, santé et soin social) entre les différents acteurs évoluant dans le processus de santé à l'école afin de fournir des interventions de soins personnalisées.
- [50] propose un système intelligent à base d'ontologie pour le contrôle de la consommation d'énergie au sein d'un campus. L'article parle d'une méthode de fusion d'informations situationnelles multidimensionnelles, qui peut être utilisée pour normaliser (assurer la cohérence des données), analyser et prédire les données multi-sources telles que les données naturelles, humanistes et spatio-temporelles sur le campus afin de faciliter les prises de décision. Dans le processus de raisonnement, la méthode de prédiction KM-KNN (KMeans – KNearest Neighbor) est utilisée pour prédire les tendances situationnelles.
- [28] présente un système à base d'ontologie pour la recommandation de contenus et de ressources aux étudiants dans un campus intelligent. Dans ce travail, un modèle d'intégration du contexte, à l'aide des ontologies, des informations de domaine et des bases de données relationnelles, a été présenté. Ainsi, il est possible d'utiliser des requêtes qui filtrent l'information selon le contexte sur la base de données relationnel.
- Le travail de [51] concerne un Framework pour la reconfiguration des applications IoT en fonction du changement de contexte dans le domaine d'un campus intelligent. Le système permet d'analyser les données (présence de personne, niveau de bruit, température intérieur, luminosité, ...) à l'aide d'un modèle basé ontologie et d'un raisonnement Bayésien. A partir de ses données le système peut déclencher la

reconfiguration pour contrôler la température, la ventilation et l'éclairage de chaque salle. L'ontologie proposée réutilise l'ontologie SSN.

- Le papier [52] traite un système intelligent à base d'ontologie pour automatiser les activités de l'enseignement supérieur. Le système utilise les techniques du raisonnement ontologique et de l'analyse de données. Les auteurs présentent la conception et l'implémentation d'une ontologie pour l'éducation. Pour l'évaluation de l'ontologie ils l'appliquent au cas de l'affectation des cours à l'enseignement le plus qualifié.
- Le système à base d'ontologie et RFID de [53] permet de surveiller les activités des étudiants sur le campus. Le RFID est utilisé pour l'assiduité des étudiants. Avec l'ontologie, les activités des étudiants sont inférées en utilisant le Framework OAM (Ontology Application Management) à partir des données RFID et des données basiques de l'université.
- [54] se concentre sur un système à base d'ontologie pour la recommandation des plannings de transport dans un campus. Le système propose deux services intelligents dont le premier est de visualiser les effectifs d'étudiants dans les bâtiments. Le second service permet de recommander aux véhicules des itinéraires permettant de faciliter le déplacement des étudiants c-à-d les véhicules passent par des points où il y'a le maximum d'étudiants.
- Le papier de [55] se concentre sur un campus d'université qu'il considère comme une organisation avec des données hétérogènes. Les auteurs construisent une ontologie qui facilite la détection de corrélation entre les différents types de données. Pour la construction de l'ontologie, les données hétérogènes sont intégrées à l'aide de graphe de connaissance. Le cas d'étude du papier est de déterminer d'une part la corrélation entre les ventes à la cafétéria et le nombre de bicyclettes garées à proximité. D'une seconde part il s'agit de déterminer la corrélation entre le nombre de bicyclettes garées et la consommation d'énergie après une période de cours. Les ontologies réutilisées sont : QB, OEMA.

Après cette brève description des différents travaux liés à notre contribution, nous résumons dans le tableau une synthèse générale des différentes informations recueillies :

Numéro	Ontologie	Année	Domaine	Ontologie réutilisée	Technologie	Implémentation	Concepts Clés	Langage d'ontologie
1	Ontology-based Course Teacher Assignment within Universities [38]	2020	Gestion Intelligente	-	Ontologie	Oui	Course, Department, Syllabus, CLO, Text_Book, Topic, Evaluation, Coordinator	RDF
2	Kids 'Smart Campus Ontology to Retrieve Interest [2]	2019	Campus Intelligent	-	Ontologie Cloud IoT	Non	Object, Network, Sensing, Time interval, Events	-
3	An ontology based context-aware architecture for smart campus applications [39]	2018	Classe Intelligente Apprentissage Intelligent	IoT-O	Ontologie IoT	Oui	Interaction context, Interaction consumers, Interaction enablers, interaction Interest, Location, Device, Service, Privilège	-
4	An Ontology for an IoT-enabled Smart Parking in a University Campus [40]	2019	Parking Intelligent	Wise-IoT SSN FOAF	Ontologie IoT	Non	Parking System, Parking Zone, Parking Lot, User, Service, Location, Sensor	OWL

5	An Ontology for an IoT-enabled Smart Classroom in a University Campus [41]	2019	Classe Intelligente	Context Ontology ONGSC FOAF SSN	Ontologie IoT	Non	User, Classroom, Service, Activity, Context, Sensor	OWL
6	A Case Study on Linked Data for University Courses [42]	2017	Gestion Intelligente	AIISO OWL Time Dublin Core FOAF	Ontologie	Oui	Course, Subject, Curriculum, Study Programme, Degree, Teacher, Location	OWL
7	Multi-Agent Systems and Ontology for Supporting Management System in Smart School[43]	2018	Gestion Intelligente	-	Ontologie Système Multi-Agent IoT [Mobile]	Oui	Person, Location, Presence, Classroom, Lesson, Evaluation, Objective	-
8	An Ontological Approach for Semantic Modeling of Curriculum and Syllabus in Higher Education[44]	2016	Gestion Intelligente	-	Ontologie	Non	Discipline, BodyOfKnowledge, KnowledgeArea, UnitOfKnowledge, Course, Syllabus, Instructor, Learning Objective, Assignment, ExamQuiz	-

9	Towards an Ontology for Representing a Student ' s Profile in Adaptive Gamified Learning System[45]	2020	Apprentissage Intelligent	-	Ontologie	Oui	Profile, Personal Information, Academic Information, Player Information, Learning style, Player type, Cognitive, Skills	OWL
10	Ontology-Based System for Educational Program Counseling [46]	2021	Gestion Intelligente	-	Ontologie	Oui	Universities, Knowledge Area, Programme, DeptProgramme, City, Country, Type	OWL
11	Developing Educational Ontology: A Case Study in Physics [47]	2018	Apprentissage Intelligent	-	Ontologie	Oui	TeachingMaterial, PersonalKnowledge, Tutorial	OWL2
12	A Framework of Decision Support System based on Integrated Data for Electricity Management in Campus[48]	2017	Gestion Intelligente	-	Ontologie IoT [RFID Sensor, Push sensor, Barcode reader, WIFI login, Voltage Sensor]	Oui	Activity, RFID, Course, Student, Construction	OWL
13	Ontology-based recommender system in higher education[49]	2019	Gestion Intelligente	-	Ontologie Machine Learning	Oui	Student, Employment, Higher Education Institution	OWL

14	An Effective Knowledge-Based Modeling Approach towards a “Smart-School Care Coordination System”[25]	2020	Santé Intelligente	-	Ontologie	Oui	Activity, Actor, Diagnosis, EHCP, Expertise, Institution, Interventions, Location	OWL
15	Multidimensional situational information fusion method for energy saving on campus[50]	2020	Gestion Intelligente	-	Ontologie Machine Learning IoT [Temperature Sensor, Brightness Sensor, Soil Moisture Meter, Electricity Meter, Flow Velocity Sensor]	Oui	Space, Time, Device, Humanistic, natural	-
16	Domain Content Querying Using Ontology-Based Context-Awareness in Information Systems[28]	2018	Apprentissage Intelligent	-	Ontologie IoT	Oui	User, Profile, Device, Learning Objects, Area, Service	OWL-DL
17	An IoT Reconfiguration Framework Applied Ontology-	2019	Gestion Intelligente	SSN	Ontologie	Oui	-	-

	based Modeling and Bayesian-based Reasoning for Context Management[51]				Machine Learning IoT [Light Sensor, Weather Sensor]			
18	Ontology-Based Smart System to Automate Higher Education Activities[52]	2021	Gestion Intelligente	-	Ontologie	Oui	Person, Course, Keywords, Publications, Experience_Since	RDF
19	Inferring Students 'Activity Using RFID and Ontology[53]	2016	Gestion Intelligente	-	Ontologie IoT [RFID Sensor]	Oui	Activity, RFID, Course, Student, Construction	OWL
20	A Recommender of Transportation Planning in Campus Using Ontology [54]	2017	Gestion Intelligente	-	Ontologie Google Maps IoT	Oui	Person, Transportation, Construction, Vehicule, Location, Main Campus	OWL
21	Ontology-Based Correlation Detection Among Heterogeneous Data Sets [55]	2020	Campus Intelligent	QB OEMA	Ontologie Google Maps IoT	Oui	University, Event, Bicycle, Consumption, Facility, Building, Parking, Activity, Agent	RDF

					[Location Sensor]			
--	--	--	--	--	-------------------	--	--	--

Tableau 2.1: Synthèse des travaux connexes

D'après la description ci-dessus des travaux connexes sur l'utilisation des ontologies dans le domaine du smart campus, nous avons pu mettre en évidence que la plupart des études n'ont pas donné une représentation complète du smart campus dans tous les aspects liés à ses sous-domaines. La plupart s'est concentrée sur une ou plusieurs sous-domaines du smart campus et n'a pas couvert tous les sous-domaines en même temps. Selon le tableau 1, nous pouvons remarquer que le sous-domaine de la Gestion intelligente était le plus utilisé. La source des données varie d'une étude à l'autre, et la majorité des études utilisent des capteurs, des données académiques, et les interactions des apprenants pour les systèmes d'apprentissage en ligne. Beaucoup de travaux ne réutilisent pas d'ontologies existants, ils ont construit l'ontologie à partir de zéro.

La plupart des travaux ne se concentrent pas suffisamment sur la suivie des activités et des enseignants afin d'utiliser ses données pour améliorer le processus d'apprentissage, ils se concentrent plutôt sur l'affectation des cours, la gestion de la consommation d'électricité ou encore la recommandation des transports. De plus, les différentes ontologies décrites ci-dessus ne présentent pas de manière claire la structure administrative d'un campus intelligent, ni une description complète de concepts telles que les cours (avec les travaux dirigés et pratiques), les bâtiments, et les différents acteurs.

Dans ce contexte, nous avons proposé une ontologie pour la gestion du campus intelligent en utilisant une approche modulaire.

Conclusion :

Nous avons essayé d'effectuer une synthèse globale des travaux en prenant en compte plusieurs critères. Enfin nous avons évoqué quelques limites de ses travaux. Le chapitre suivant décrit en détails la conception.

Chapitre 3 : Conception

Introduction :

Notre travail est une amélioration de notre système de gestion de la pédagogie au sein de l'université 8 Mai 1945. Le point le plus important de ses améliorations est le système de génération automatique de l'emploi de temps à l'aide de l'utilisation de l'algorithme génétique. L'emploi du temps est un concept très important dans le campus intelligent car il est utilisé dans la gestion de la pédagogie, la scolarité, classe intelligente, paiement des enseignants (pointage). C'est une tâche primordiale pour le suivie des activités des étudiants et des enseignants au sein du campus intelligent. Ainsi la gestion de la pédagogie (incluant la gestion de l'emploi du temps) est le noyau de notre architecture pour l'élaboration du campus intelligent. Dans ce chapitre nous faisons la présentation des deux contributions (i) la génération de l'emploi du temps (ii) Un système de gestion d'un campus intelligent basée sur une ontologie modulaire et une architecture Cloud-Fog-Edge

1 Première contribution : La génération de l'emploi du temps :

La gestion des emplois du temps consiste à affecter les enseignants, les étudiants et les salles à des cours dans l'espace de temps tout en satisfaisant un ensemble de contraintes. Avec les outils traditionnels, c'est une tâche fastidieuse et répétitive. Nous avons proposé un algorithme pour automatiser la création de l'emploi. Nous avons utilisé l'algorithme génétique[56] car il permet de réduire le temps nécessaire pour générer l'emploi du temps avec une plus grande précision et moins d'erreur [4].

1.1 Les contraintes pour la génération de l'emploi du temps :

Pour l'obtention d'un emploi du temps de qualité, ils existent un certain nombre de contraintes à respecter. On les a classés en deux catégories :

- Les contraintes dures :
 - A un instant donné, un enseignant ne peut assurer qu'un seul enseignant
 - A un instant donné, une salle ne peut être occupé que pour un seul enseignant
 - A un instant donné, un groupe ne peut assister qu'a un seul enseignant

- Un enseignement ne peut pas avoir lieu que durant les horaires permis
- On ne peut pas mettre des étudiants dans des salles où il y'a pas assez de places
- Un groupe doit avoir obligatoirement un temps libre pour l'heure du déjeuner
- Un TP ne peut avoir lieu que dans une salle de TP ou un laboratoire
- Un groupe ou un enseignant ne peut pas avoir de trois (3) séances successives
- Les contraintes souples :
 - Satisfaire le vœu de l'enseignant
 - Eviter les séances creuses
 - Eviter de placer les cours l'après-midi
 - L'enseignant doit être présent au moins 3 fois par semaine au département
 - Homogénéiser l'emploi du temps de l'enseignant

1.2 Structure de donnée :

La première étape de notre travail est de définir les données qui font l'objet de traitement et de manipulation. Pour le problème de gestion de l'emploi du temps, on a les données suivantes :

- La liste des modules
- La liste des enseignants
- La liste des salles
- La liste des jours
- La liste des horaires
- La liste des groupes
- La liste des affectations
- La liste des vœux

A partir de ces données, on construit une liste des séances de l'emploi de temps. Cette liste est construite de telle sorte que chaque séance d'un groupe occupe un seul horaire.

1.3 Le codage :

Nous proposons le codage suivant :

Le chromosome est un tableau de taille (m * g) avec m le nombre de module et g le nombre de groupe. Les éléments du tableau (Gènes) sont représentés par la combinaison **<Module, Groupe, Jour, Horaire, Salle, Enseignant>**. Le chromosome est représenté en utilisant un codage réel. Chaque composante du gène dans le chromosome est représentée par son identifiant c'est-à-dire le module compilation sera représenté par 23 (son identifiant dans la liste de module), l'enseignant Hamadoun Cisse par 145 (son identifiant dans la liste des enseignants).

Par exemple pour le Gène **<Génie Logiciel, Groupe 2, Lundi, 08:00-09:30, E8.9, Lynda Djakhdjakha>** sera **<23, 2, 2, 1, 9, 26 >**.

Le chromosome est représenté par :

<Module 1, Groupe 1, Jour 1, horaire 1, Salle 1, Enseignant 1 >	<Module 2, Groupe 2, Jour 1, horaire 1, Salle 2, Enseignant 2 >
---	---	-------	-------

Tableau 3.1 : Représentation du chromosome

1.4 Population initiale :

La population initiale est une matrice formée par n chromosomes qui sont créés de manière aléatoire. Le choix du n est une tâche très délicate car n trop grand ralentit la vitesse de l'algorithme et n trop petit nous donne des solutions pas vraiment optimales [57].

L'algorithme de génération de la population initiale est le suivant :

Algorithme Initialisation population

Debut

Pour chaque individu

 Pour chaque groupe

 Pour chaque cours

 Sélectionner aléatoirement un jour

 Sélectionner aléatoirement un horaire

 Sélectionner aléatoirement une salle

Sélectionner aléatoirement un enseignant
Créer une séance avec le module, groupe, module, jour, horaire, salle, enseignant
Finpour
Finpour
Finpour
Fin

Algorithme 1 : Initialisation de la population

1.5 La fonction objective :

Dans notre algorithme, nous avons décidé de travailler avec deux fonctions objectives. La valeur de la première fonction est calculée selon la satisfaction des contraintes dures. La valeur de la seconde est calculée selon les contraintes souples. Ce choix est effectué pour notre besoin d'avoir un emploi du temps dans le pire des cas où les contraintes dures ont été satisfaites mais pas toutes contraintes souples. Avec une seule fonction objective cela ne sera pas possible car on ne pourra pas savoir si l'algorithme n'a pas convergé à cause des contraintes souples

La fonction objective est représentée mathématiquement par la formule :

$$\text{Minimiser } \sum_{i=0}^n f_i$$

avec :

n : le nombre de contrainte

f : fonction d'évaluation de la contrainte

1.6 L'algorithme de sélection :

Parmi les différentes méthodes existantes, nous avons décidé la sélection par tournoi [58]. Elle consiste à, chaque fois que l'on veut choisir un individu, à sélectionner un sous-ensemble de la population et à ne conserver que le meilleur élément du sous-ensemble. On a effectué cette méthode car elle augmente les chances pour les individus de piètre qualité de participer à l'amélioration de la population et il est très rapide à implémenter [59].

1.7 L'opération de croisement :

Le croisement consiste à choisir aléatoirement un point de croisement pour chaque couple de chromosome. Dans notre cas, nous avons opté pour un croisement uniforme. Chaque bit est

choisi parmi les parents avec la même probabilité. Ce choix se justifie par le fait que le croisement uniforme est la meilleure option pour la génération de l'emploi du temps [3].

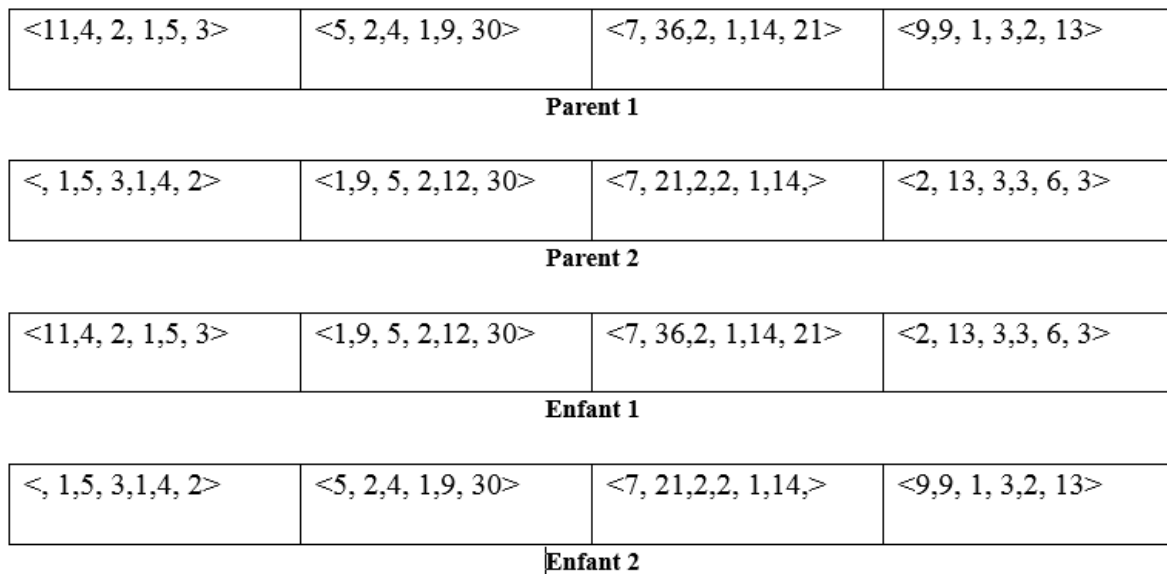


Figure 3.1 : Illustration de l'opération de croisement

1.8 L'opération de mutation :

Une mutation est une inversion d'un bit aléatoire dans un chromosome. Dans notre cas, elle consiste à changer certaines séances de notre chromosome choisi aléatoirement.

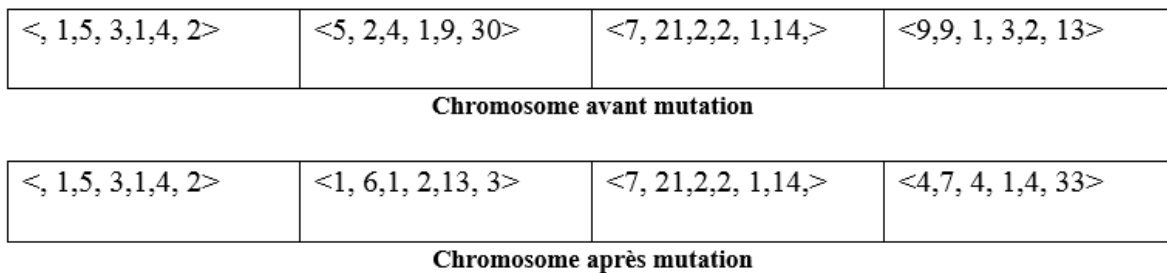


Figure 3.2 : Illustration de l'opération de mutation

1.9 Critères d'arrêt :

Pour l'arrêt de notre algorithme nous avons proposé deux critères. Nous avons fixé un certain nombre de générations (itérations). Si l'algorithme converge avant la fin du nombre de

génération, alors l'algorithme s'arrête. Dans le cas l'algorithme ne converge pas, elle devra s'arrêter après atteint le nombre d'itération.

Nous avons dans cette section présenté la première contribution de ce mémoire. Nous avons décrit les étapes de base de l'algorithme génétique choisi pour la génération de l'emploi du temps. Nous avons ainsi défini deux fonctions objectives basées sur l'ensemble de contraintes listées. L'implémentation de l'algorithme fera l'objet du chapitre suivant.

2 Deuxième contribution : La gestion d'un campus intelligent basée sur une ontologie modulaire et architecture Cloud-Fog-Edge :

La deuxième contribution dans ce mémoire est la proposition d'un système de gestion d'un campus intelligent. Notre système inclue plusieurs modules répartis sur trois niveaux (Cloud, Fog et Edge).

2.1 Conception de l'Ontologie :

Pour le développement de notre ontologie ,nous avons utilisé la méthodologie de conception **NeOn** [60] car c'est la méthode la plus complète pour le développement d'ontologie modulaire [61].

2.1.1 Les modules de l'ontologie campus intelligent :

L'ontologie du campus intelligent comporte douze (12) modules couvrant l'ensemble des sous domaines du campus intelligent décrits dans la section Concept de base de l'état de l'art. Seul le sous-domaine de la santé intelligente n'est pas couvert à cause du manque d'information. Les modules du campus intelligent sont :

- **Module « Acteur »** : Ce module décrit les différents acteurs du campus intelligent. Il décrit l'étudiant, l'enseignant, et l'ensemble des employés du campus. Il permet la présentation des différents acteurs avec leur caractéristique. Le module peut être utilisé dans les ontologies du campus intelligent où le concept acteur (Personne) est requis.

- **Module « Capteur »** : Ce module décrit les différents capteurs déployés au sein d'un campus intelligent. Elle présente une hiérarchie complète des capteurs avec leurs familles. Ce module peut être utilisé par tout système intelligent.

- **Module « Outils »** : ce module décrit l'ensemble des outils pouvant être commandés de manière automatique au sein d'un campus. Parmi ces outils, on distingue les lampes, les portes, les climatiseurs, etc. Il peut aussi être utilisé pour la conception de système intelligent.
- **Module « Enseignement »** : Ce module décrit l'ensemble des unités d'enseignement au sein d'un campus intelligent. Il présente une hiérarchisation complète des différents concepts dans le domaine de l'enseignement. Il est utilisable dans les systèmes pour la gestion de l'apprentissage.
- **Module « Batiment »** : Ce module fait une description complète des bâtiments au sein du campus intelligent. Il permet de spécifier le type de bâtiment et sa position. Il peut être utilisé à la base de tout système nécessitant l'usage d'un bâtiment.
- **Module « Organisation »** : Ce module décrit l'ensemble des structures au sein du campus intelligent. Il présente l'ensemble des organismes allant du groupe auquel appartient l'étudiant jusqu'au rectorat où siège le recteur.
- **Module « Activité »** : Ce module définit l'ensemble des activités pédagogiques, administratives, sociales culturelles au sein du campus intelligent.
- **Module « Emploi du temps »** : Ce module est une description complète de concepts de la tâche emploi du temps. Il décrit principalement les dépendances entre les différents concepts collaborant pour la réalisation de l'emploi du temps.
- **Module « Affectation »** : Il définit de façon claire et détaillée les concepts participant dans le processus d'affectation d'unité d'enseignement à un enseignant. Il permet de décrire les compétences pédagogique et scientifiques associés à chaque enseignant.
- **Module « Jardin »** : Ce module définit les concepts de la gestion du jardin. Il présente l'acteur, les outils, les capteurs utilisées pour gérer de manière intelligente l'irrigation des plantes au sein des jardins du campus intelligent.
- **Module « Classe intelligente »** : Ce module décrit l'ensemble de concepts du processus de gestion d'une classe intelligente. Il permet de découvrir les capteurs et les outils qui font l'intelligence de la classe. Il permet aussi d'associer à chaque classe son activité. Il peut être à la base de tout système dans le domaine du campus intelligent.
- **Module « Parking »** : Ce module est une description complète des concepts de la tâche de gestion du parking intelligent. Il permet de spécifier l'ensemble des composants (outils et composants) et les concepts du processus de réservation de places.

2.1.2 Structuration et développement des modules du campus intelligent :

2.1.2.1 Structuration :

Une ontologie modulaire est constituée de deux modules principaux : un module noyau et les modules thématiques [62], [63]. Pour notre ontologie modulaire, le module noyau est le module « Acteur » et les modules thématiques sont les suivants : module « Outils », module « Capteur », module « Organisation », module « Enseignement », module « Emploi du temps », module « Parking », module « Batiment », module « Activité », module « Classe intelligente », module « Affectation », et le module « Jardin ».

Les différents modules de l'ontologie sont liés par des relations d'interconnexion permettant d'interroger et de raisonner sur l'ontologie globale.

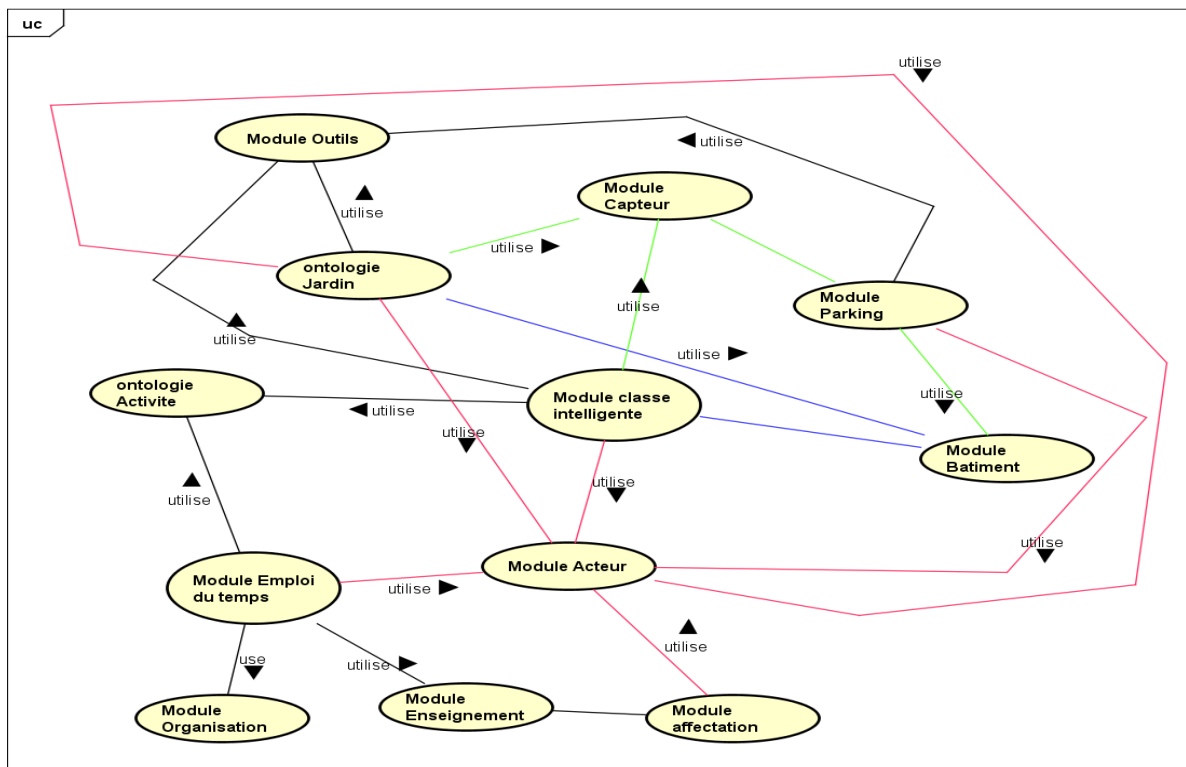


Figure 3.3 : Structure des modules

2.1.2.2 Développement des modules :

Dans ce mémoire, nous décrivons les étapes de développement de deux (2) modules. Il s'agit du module noyau « Acteur » et du module thématique « Classe Intelligente ».

2.1.2.2.1 Le module « Acteur » :

- **Objectif et Spécification des besoins :**

Les objectifs de ce module sont :

- La représentation des acteurs du campus intelligent
- ✓ Offrir une description utile et complète des acteurs intervenant dans le campus intelligent en vue de faciliter leur utilisation dans les processus de gestion.
- ✓ Définir une hiérarchie complète permettant la classification des différents acteurs.

- **Acquisition de connaissances :**

Pour l'élaboration de notre ontologie nous avons utilisé beaucoup de ressources pour l'acquisition des connaissances du domaine à travers une recherche sur les structures des universités. Nous présenterons ses différents concepts dans l'étape suivante dans un tableau récapitulatif.

- **Conceptualisation :**

Cette phase prend en entrée la liste des concepts acquis dans la phase précédente. Pour visualiser la hiérarchisation de ces concepts, nous utilisons le diagramme de classe d'UML. La figure suivante représente les différents concepts liés au premier module.

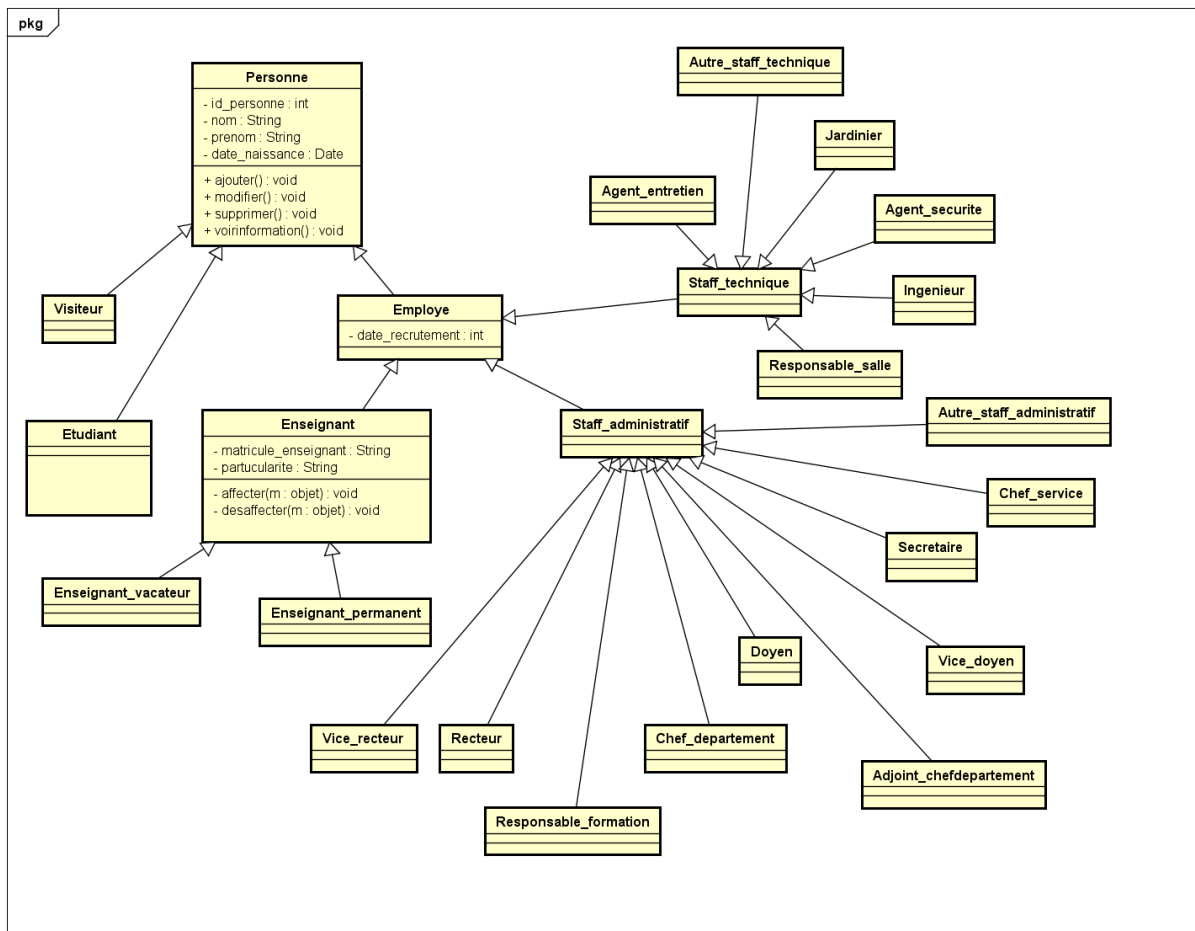


Figure 3.4 : Diagramme de classe Module « Acteur »

Ainsi le tableau suivant présente l'ensemble des concepts, des relations et des attributs.

Name	Description	Type
Personne	Fait référence à la personne	Concept
Visiteur	Fait référence au visiteur du campus	Concept
Etudiant	Fait référence à l'étudiant	Concept
Employe	Fait référence à l'employé	Concept
Enseignant	Fait référence à l'enseignant	Concept

Enseignant_vacataire	Fait référence à l'enseignant vacataire	Concept
Enseignant_permanent	Fait référence à l'enseignement permanent	Concept
Staff_administratif	Fait référence à l'employé du staff administratif	Concept
Staff_technique	Fait référence à l'employé du staff technique	Concept
Recteur	Fait référence au Recteur	Concept
Vice_recteur	Fait référence au Vice-recteur	Concept
Doyen	Fait référence au Doyen	Concept
Vice-doyen	Fait référence au Vice-doyen	Concept
Chef_service	Fait référence au chef de service	Concept
Chef_departement	Fait référence au chef de département	Concept
Adjoint_chefdepartement	Fait référence à l'adjoint du chef de département	Concept
Responsable_formation	Fait référence au responsable de formation	Concept
Secrtaire	Fait référence au secrétaire	Concept
Autre_staff_administratif	Fait référence à l'autre personne travaillant dans le staff administratif	Concept

Staff_technique	Fait référence à l'employé du Staff technique	Concept
Agent_entretien	Fait référence à l'agent d'entretien	Concept
Responsable_salle	Fait référence à la responsable de salle	Concept
Ingenieur	Fait référence à l'ingénieur	Concept
Agent_securite	Fait référence à l'agent de sécurité	Concept
Jardinier	Fait référence au jardinier	Concept
Autre_staff_technique	Fait référence à l'autre personne travaillant dans le staff technique	Concept
nom	Fait référence au nom de la personne	Attribut
prenom	Fait référence au prénom de la personne	Attribut
date_naissance	Fait référence à la date naissance de la personne	Attribut
id_personne	Fait référence au numéro d'identification nationale de la personne	Attribut
date_recrutement	Fait référence à la date de recrutement de l'employé	Attribut

particularite	Fait référence à la particularité de l'enseignant	Attribut
matricule_etudiant	Fait référence à l'identifiant de l'étudiant au sein du campus	Attribut
matricule_enseignant	Fait référence à l'identifiant de l'enseignant au sein du campus	Attribut
matricule_staff	Fait référence à l'identifiant des employés des staffs au sein du campus	Attribut

Tableau 3.2 : Liste des concepts, relations, attributs Module « Acteur »

Les phases formalisation et implémentation feront l'objet du chapitre implémentation.

2.1.2.2.2 Le module « Classe intelligente » :

- **Objectif et Spécification des besoins :**

Les objectifs principaux de ce module sont :

- La gestion de classe intelligente.
- Offrir une description utile et complète du processus de la classe intelligente.
- Définir une méthodologie de gestion intelligente en vue d'améliorer le processus d'apprentissage.
- ✓ Permettre la réduction de la consommation au sein de la classe.

- **Acquisition de connaissances :**

Pour l'élaboration de cette ontologie, les connaissances du domaine ont été obtenues grâce aux travaux de recherche du domaine et aussi grâce à un expert du domaine. Les connaissances du domaine et les concepts clés seront présentés dans l'étape suivante.

- **Conceptualisation :**

Le tableau et la figure suivants présentent l'ensemble des concepts, des relations et des attributs.

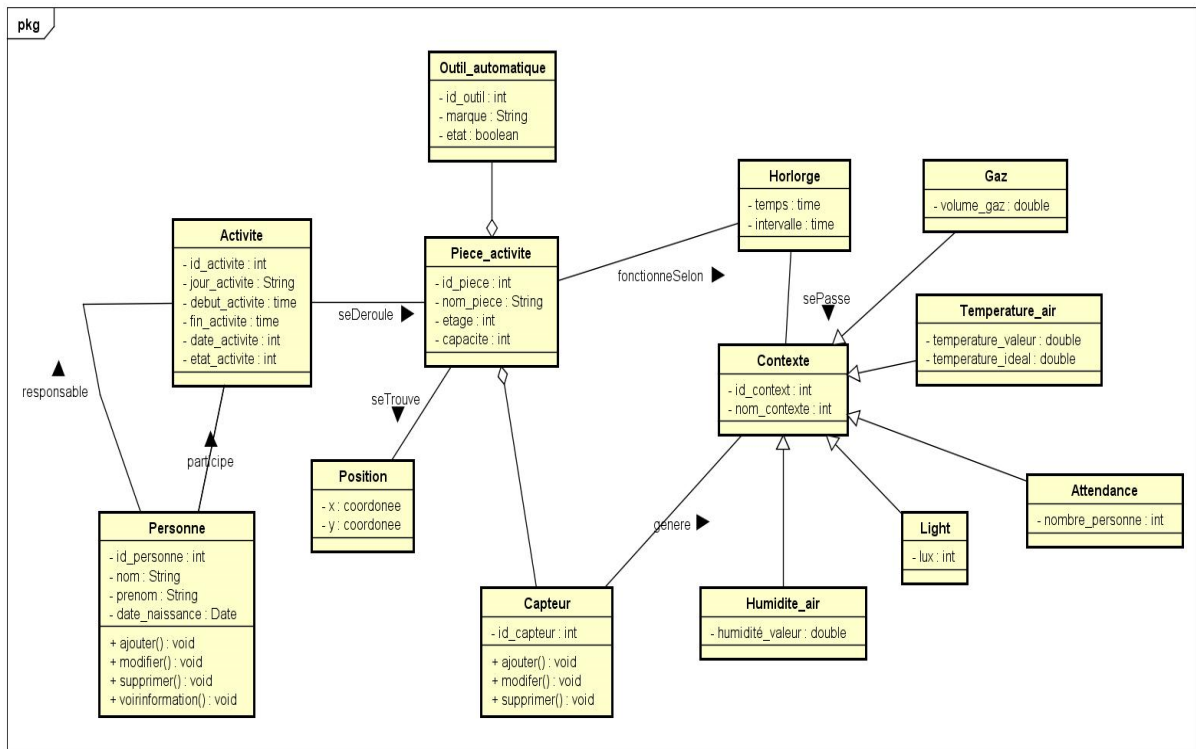


Figure 3.5 : Diagramme de classe du module « Classe Intelligente »

Concept	Description	Type	Module Réutilisé
Personne	Fait référence à la personne	Concept	Personne
Activite	Fait référence à l'activité pour laquelle la salle est reservé	Concept	Activite
Position	Fait référence à la localisation de la salle	Concept	
Piece_activite	Fait reference à la salle	Concept	Batiment
Capteur	Fait référence au capteur utilisé dans la salle	Concept	Capteur
Horloge	Fait référence à l'horloge utilisé dans la salle	Concept	

Contexte	Fait référence au contexte acquis grâce au capteur	Concept	
Outils_automatique	Fait référence au outils automatique utilisé dans la salle	Concept	Outils
Gaz	Fait référence au contexte gaz	Concept	
Temperature_air	Fait référence au contexte température	Concept	
Humidite_air	Fait référence au contexte humidité	Concept	
Light	Fait référence au contexte luminosité	Concept	
Attendance	Fait référence au contexte attendance	Concept	
participe	Fait référence à la participation de la personne à une activité	Relation	
responsable	Fait référence à la prise en charge d'une activité par une personne	Relation	
seTrouve	Fait référence à la localisation de la salle à une position	Relation	
seDeroule	Fait référence au déroulement d'une activité dans une salle	Relation	
genere	Fait référence à la génération de contexte par les capteurs	Relation	

fonctionneSelon	Fait référence au fonctionnement de la salle selon à un instant	Relation	
sePasse	Fait référence au contexte émis à un instant précis	Relation	
nom	Fait référence au nom de la personne	Attribut	
....

Tableau 3.3 : Listes des concepts, relations et attributs du Module « Classe Intelligente »

Les phases formalisation et implémentation feront l'objet du chapitre implémentation.

2.2 Architecture du Système :

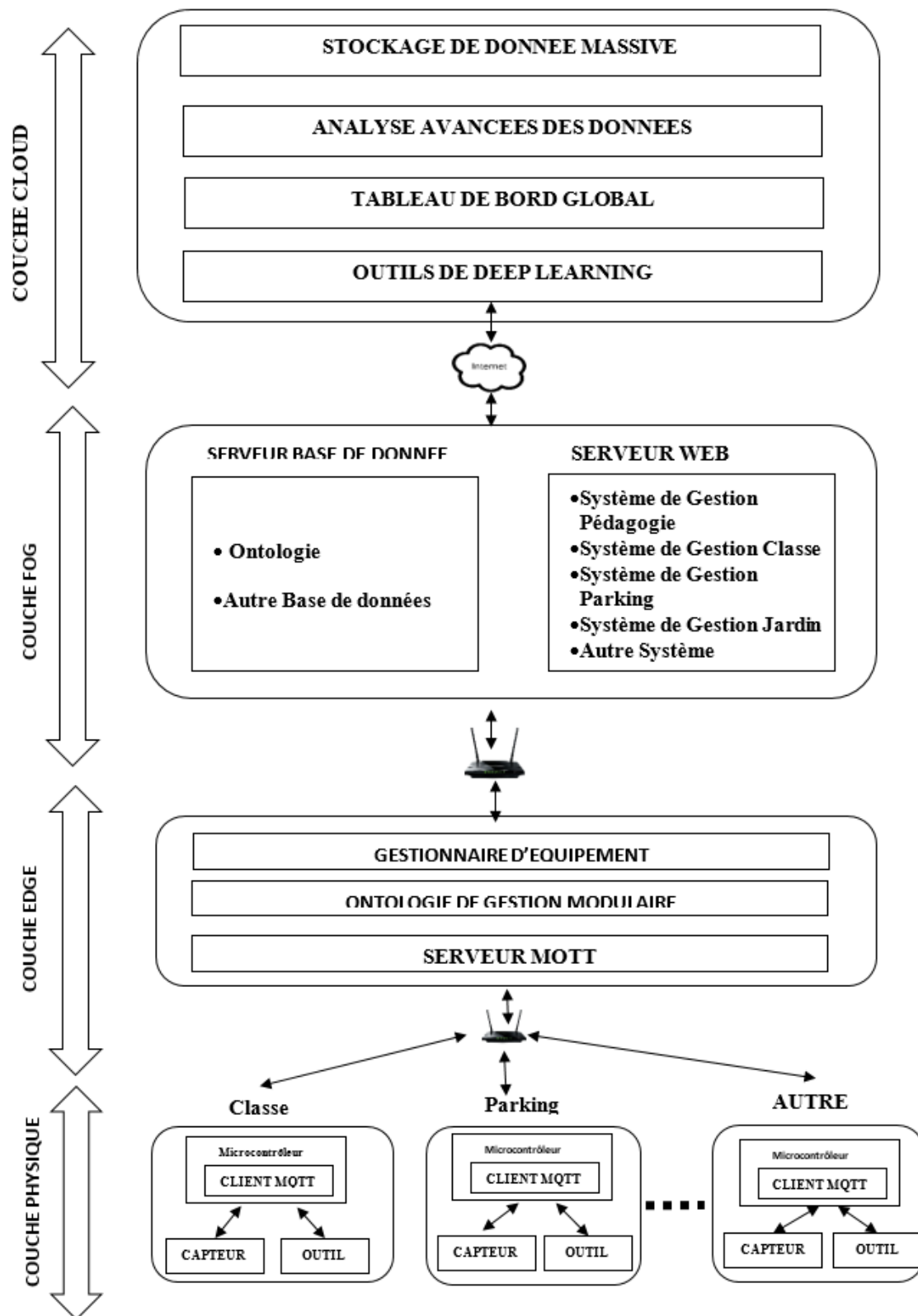


Figure 3.6 : Architecture proposée

Nous avons utilisé une architecture Cloud, Fog, Edge pour la conception de notre système.

- Le niveau Cloud : Le Cloud est utilisé pour le stockage permanent de nos données massives, et les analyses qui en découlent. Il permet aux acteurs qualifiés de visualiser les tableaux de bord relatif aux données. Il offre aussi le moyen d'implémenter nos outils d'apprentissage profond (Deep Learning) nécessitant de grandes puissances de calcul.

Les applications de gestion sont stockées sur le Fog car ils nécessitent un temps de latence moyen, c'est-à-dire pas trop rapide pour être mis au niveau du Edge mais nécessitant plus de puissance de calcul que celui-ci.

- Le niveau Fog : L'architecture Fog fournit les outils nécessaires pour le traitement des données des applications de gestion, telles que la gestion de la pédagogie, la gestion du parking, etc.

Les outils en temps réel tels que les ontologies de gestion modulaire (ontologie pour la gestion de salle, parking, etc.) sont stockées au niveau de l'Edge. Un retard dans la coordination des différents outils peut être fatale dans les situations où par exemple l'alarme d'incendie est déclenchée en retard à cause du temps de réponse du système.

- Le niveau Edge : L'architecture Edge permet de réduire l'utilisation de ressource, d'améliorer la gestion des tâches au sein du campus intelligent.

La couche physique est constituée de l'ensemble des microcontrôleurs, des capteurs et des outils contrôles grâce aux données des capteurs. C'est la couche la plus bas de notre architecture et nécessite des traitements en temps réel.

Une description du fonctionnement de l'architecture sera présentée dans le chapitre implémentation.

Conclusion :

Dans ce chapitre, nous avons décrit les étapes de notre algorithme de génération de l'emploi du temps.

Nous avons aussi présenté l'architecture de notre système ainsi que les différents modules de notre ontologie modulaire ainsi que la conception du module « Acteur » et du module « Classe intelligente ».

L'implémentation de la génération de l'emploi du temps et l'ontologie se fera dans le chapitre suivant de notre mémoire.

Chapitre 4 : Implémentation

Introduction :

Dans ce chapitre, nous allons présenter l'implémentation de notre système. Nous commencerons par l'implémentation de la génération automatique de l'emploi du temps avec un scénario d'exécution pour la génération de l'emploi du temps. Ensuite, nous présenterons en détails l'implémentation des modules « Acteur » et « Classe intelligente ». Nous décrirons dans chacune des implémentations les outils utilisés.

1 Outils :

Dans cette partie nous ne parlons que des outils utilisés pour l'implémentation de l'ontologie notre système.

1.1 MQTT :

MQTT, pour "Message Queuing Telemetry Transport", est un protocole open source de messagerie qui assure des communications non permanentes entre des appareils par le transport de leurs messages [64]. Nous l'avons utilisé pour la communication entre les capteurs et le niveau Edge.

1.2 Flask :

C'est un Framework python permettant la réalisation d'un site web ou d'une api web. Son principal avantage est d'être simple à utiliser mais sans perdre de fonctionnalités, de plus il peut quasiment tout faire grâce à de nombreuses extensions [65]. On l'utilise pour construire notre api qui nous permet d'interagir avec notre ontologie à partir de requête HTTP (Hypertext Transfer Protocol).

1.3 Owready2 :

C'est un module python pour la manipulation d'ontologie [66].

2 Implémentation de la génération de l'emploi du temps :

La génération automatique de l'emploi du temps a été intégrée à notre système de gestion de la pédagogie. L'algorithme prend en entrée la liste des enseignants, des modules, des salles, des groupes, des vœux des enseignants et aussi la liste des heures de cours. L'algorithme essaye de

générer le meilleur emploi en respectant l'ensemble des contraintes relatives à la création de l'emploi du temps. La figure suivante illustre un extrait du code de l'algorithme de génération de l'emploi du temps.

```
5
6 class Schedule:
7     def __init__(self, data):
8         self._data = data
9         self._lectures = []
10        self._number_of_conflicts = 0
11        self._fitness = -1
12        self._classNumb = 0
13        self._is_fitness_changed = True
14
15    def get_lectures(self):
16        self._is_fitness_changed = True
17        return self._lectures
18
19    def get_number_of_conflicts(self):
20        return self._number_of_conflicts
21
22    def get_fitness(self):
23        if self._is_fitness_changed:
24            self._fitness = self.calculate_fitness()
25            self._is_fitness_changed = False
26        return self._fitness
27
28    def initialize(self):
29        courses = self._data.get_courses()
30        for i in range(0, len(courses)):
31            modules = courses[i].get_modules()
32            for j in range(0, len(modules)):
33                new_meeting_time = self._data.get_meeting_times()[rnd.randrange(0, len(self._data.get_meeting_times()))]
34                new_lecture = Lecture(self._classNumb, courses[i], modules[j], new_meeting_time)
35                self._classNumb += 1
36
37                new_lecture.set_room(self._data.get_rooms()[rnd.randrange(0, len(self._data.get_rooms()))])
38
39                new_lecture.set_docent(
```

Figure 4.1 : Extrait du code de l'algorithme génétique

Le code ci-dessus décrit l'initialisation de la population initiale. Une explication plus claire est présentée dans la partie conception de la génération automatique.

Pour la génération de l'emploi du temps, un ensemble d'étape est nécessaire. Les enseignants doivent remplir au préalable la liste de leurs vœux pour l'emploi du temps. Ensuite le chef de la pédagogie procède à l'exécution de l'algorithme après récupération de tous les vœux et de toutes les entrées. Les figures suivantes illustrent l'ensemble de ces étapes.

La figure ci-dessous décrit la page d'accueil de notre application de gestion de la pédagogie.

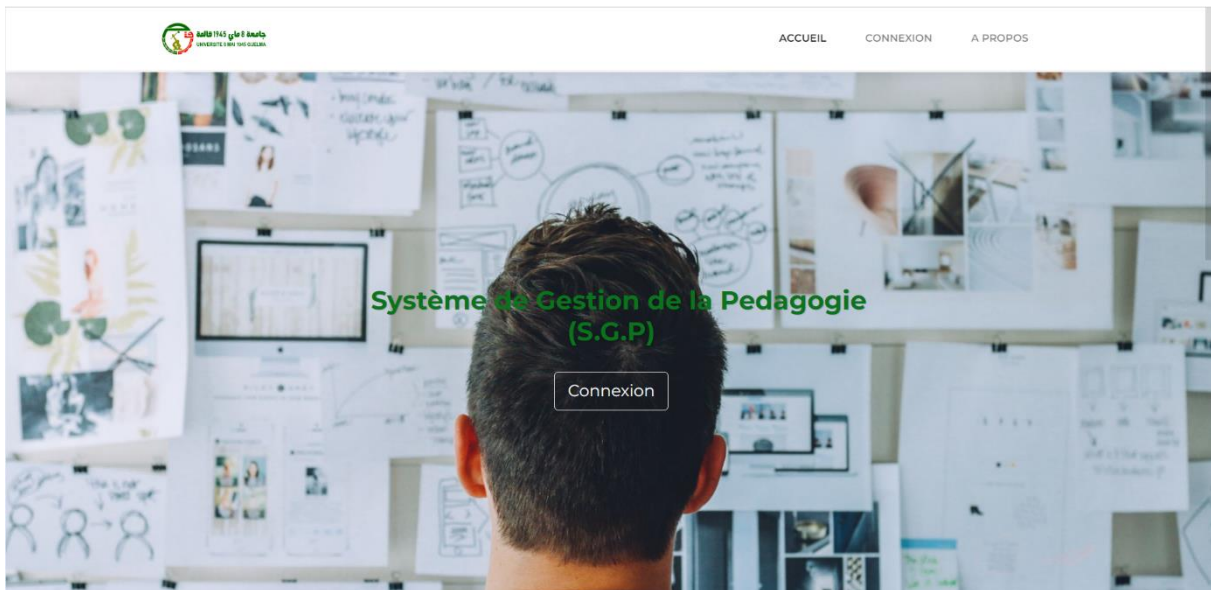


Figure 4.2 : Page d'accueil de l'application

La figure ci-dessous présente l'attention d'un enseignant dans notre système.

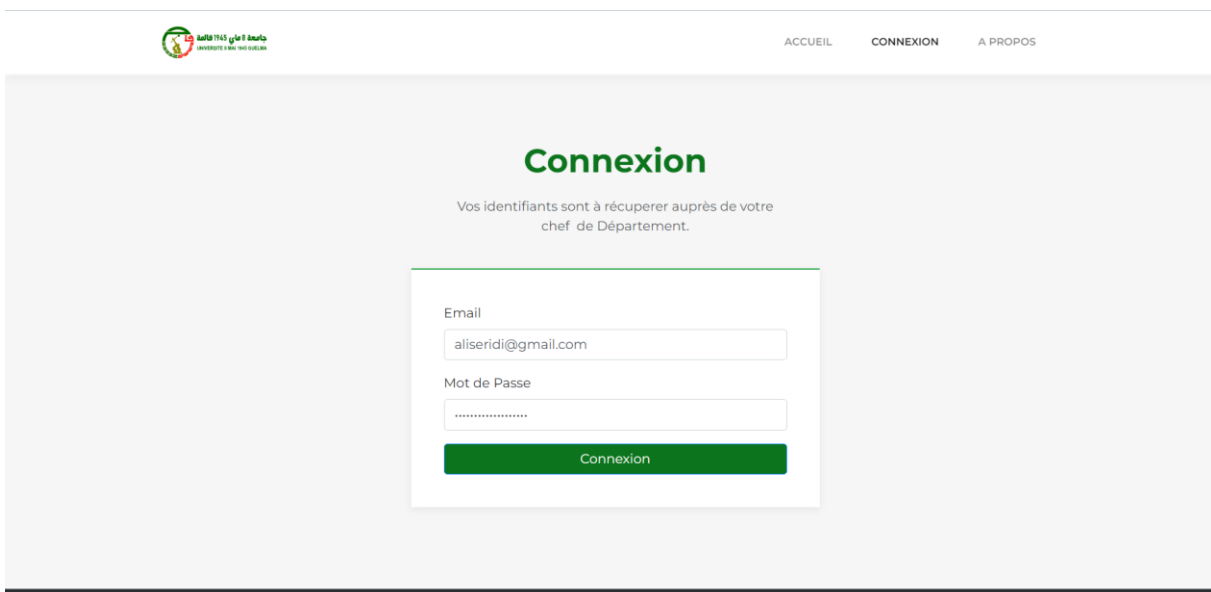


Figure 4.3 : Connexion d'un enseignant

La figure ci-dessous est la page d'accueil de l'enseignant. A partir de cette page, il peut visualiser les tableaux de bord concernant son module, ses retards, etc. Il peut accéder à toutes les fonctionnalités du système depuis sa page d'accueil.



Figure 4.4 : Page d'accueil de l'enseignant

La figure ci-dessous représente le menu de remplissage de la fiche de vœux.

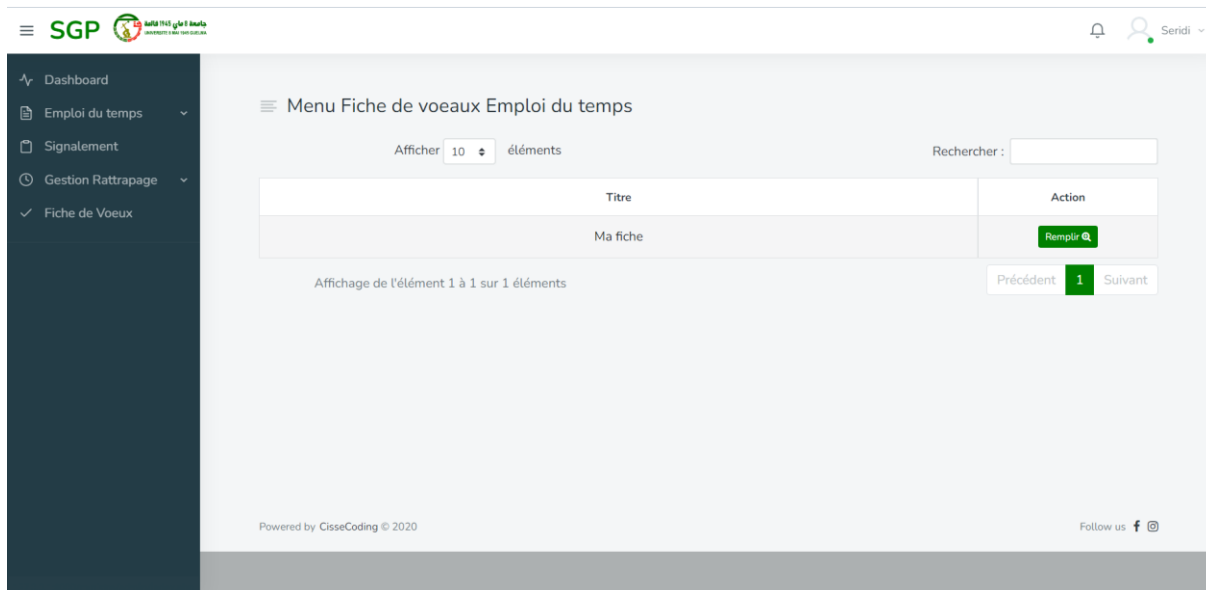


Figure 4.5 : Menu des vœux de l'enseignant

La figure ci-dessus décrit le remplissage d'une case de la fiche de vœux. L'enseignant sélectionne une case vide sur son planning de vœux et sélectionne le module qu'il compte y enseigner durant cette plage horaire.

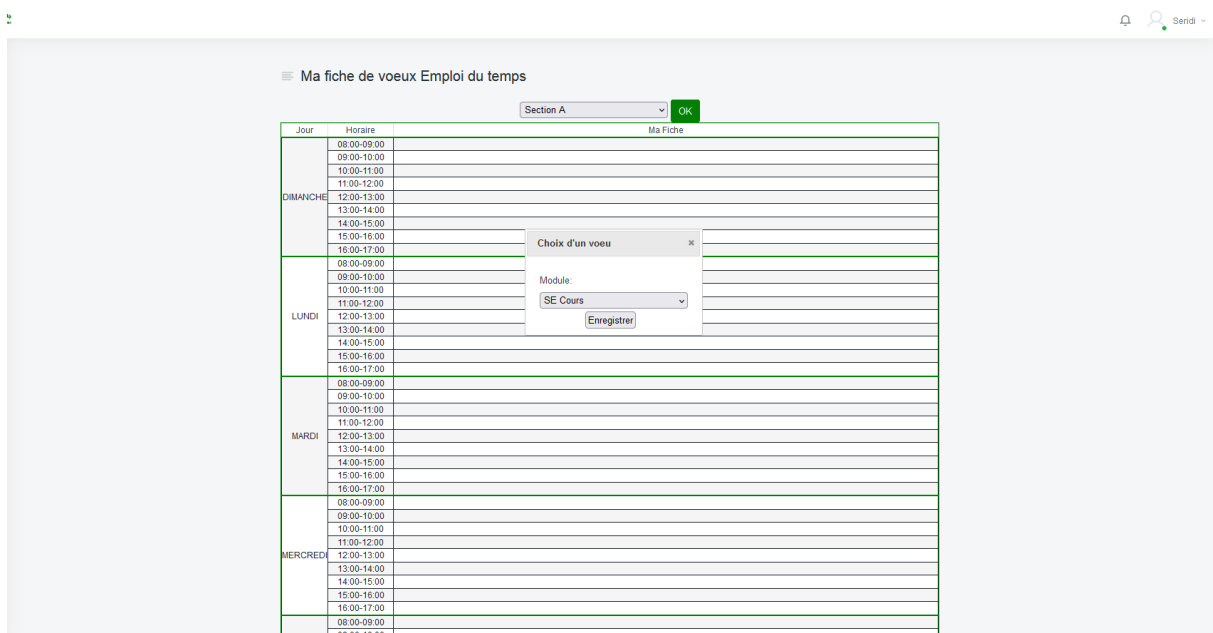


Figure 4.6 : Enregistrement d'un vœu

La figure suivante décrit la fiche de vœu après le remplissage d'un choix.

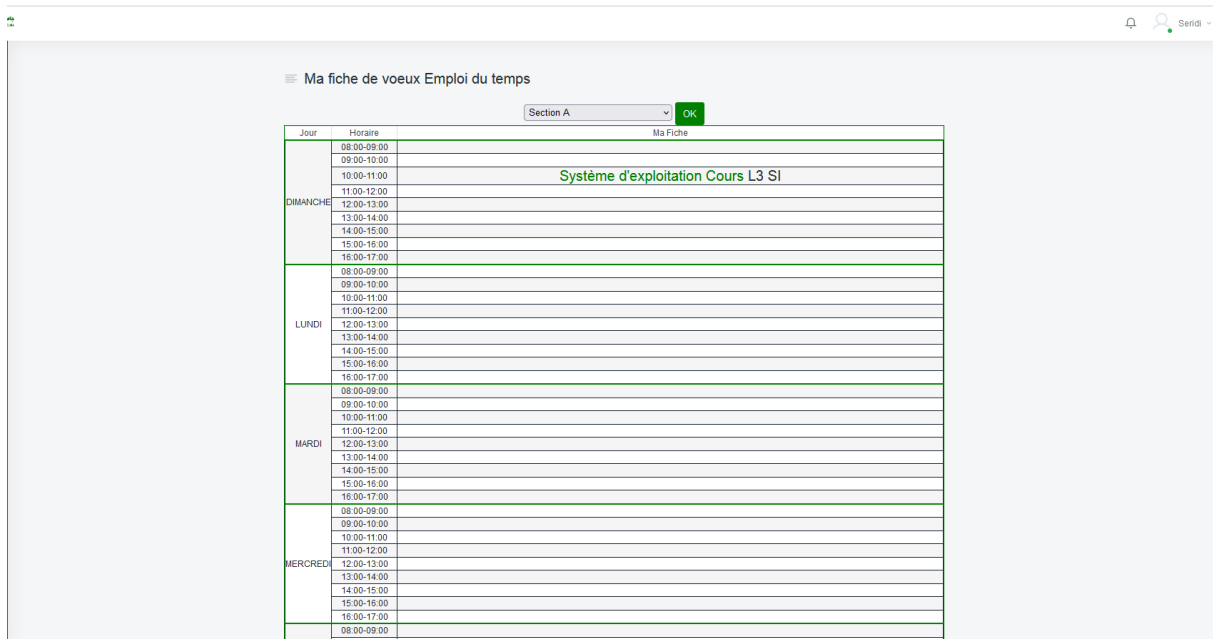



Figure 4.7 : Affichage du vœu enregistré

La figure ci-après représente le remplissage complet de la fiche de vœux par l'enseignant.

Jour	Horaire	Ma Fiche De Vœux Emploi Du Temps
DIMANCHE	08:00-09:00	
	09:00-10:00	
	10:00-11:00	
	11:00-12:00	
	12:00-13:00	
	13:00-14:00	Système d'exploitation 2 Cours L3 SI
	14:00-15:00	Système d'exploitation 2 Cours L3 SI
LUNDI	15:00-16:00	Système d'exploitation 2 TD L3 SI G1
	16:00-17:00	Système d'exploitation 2 TD L3 SI G1
	08:00-09:00	
	09:00-10:00	
	10:00-11:00	
	11:00-12:00	
	12:00-13:00	
MARDI	13:00-14:00	
	14:00-15:00	
	15:00-16:00	
	16:00-17:00	
	08:00-09:00	Système d'exploitation 2 TP L3 SI G2
	09:00-10:00	Système d'exploitation 2 TP L3 SI G2
	10:00-11:00	Système d'exploitation 2 TP L3 SI G1
MERCREDI	11:00-12:00	Système d'exploitation 2 TP L3 SI G1
	12:00-13:00	
	13:00-14:00	
	14:00-15:00	
	15:00-16:00	
	16:00-17:00	
	JEUDI	08:00-09:00
09:00-10:00		Algorithme et Structure de Données 3 TP L2 Informatique G4
10:00-11:00		Algorithme et Structure de Données 3 TP L2 Informatique G3
11:00-12:00		Algorithme et Structure de Données 3 TP L2 Informatique G3
12:00-13:00		
13:00-14:00		
14:00-15:00		

Figure 4.8 : Remplissage complet Fiche de vœux

Les figures suivantes illustrent l'intervention du chef de la pédagogie dans la génération de l'emploi du temps. La figure suivante illustre sa connexion au système.



[ACCUEIL](#) [CONNEXION](#) [A PROPOS](#)

Connexion

Vos identifiants sont à récupérer auprès de votre chef de Département.

Email

Mot de Passe

Connexion

Figure 4.9 : Connexion du chef de la pédagogie

La figure suivante illustre la page d'accueil du chef de la pédagogie. Comme pour l'enseignant, il peut visualiser les tableaux de bord mais il s'agit de ceux concernant tout le département.

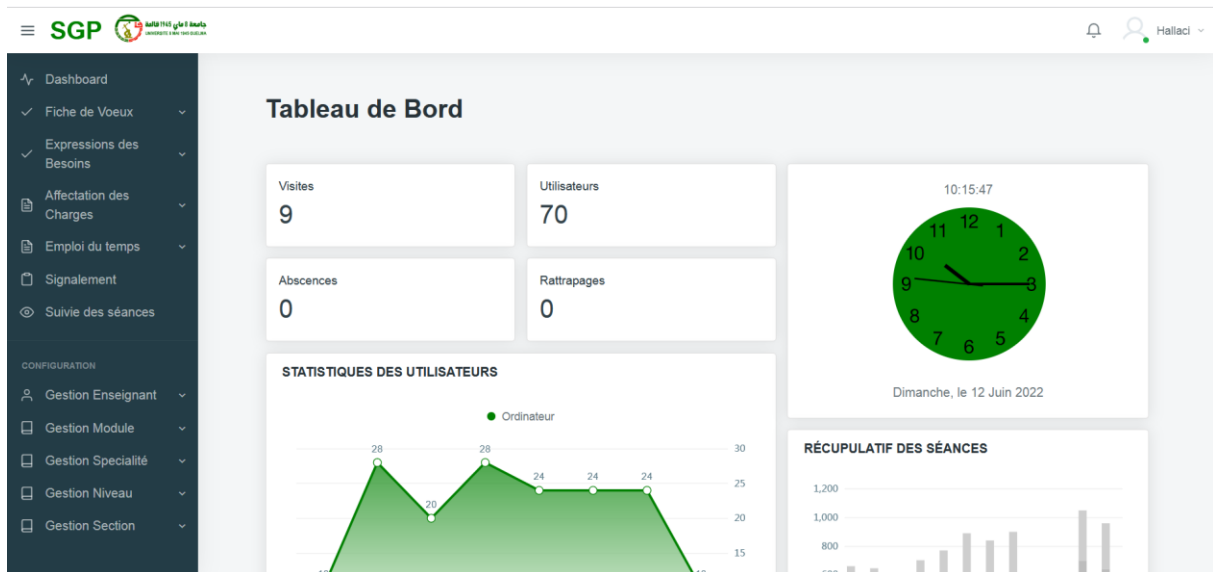


Figure 4.10 : Page d'accueil du chef de la pédagogie

La figure ci-après décrit la première étape pour la création de l'emploi du temps.

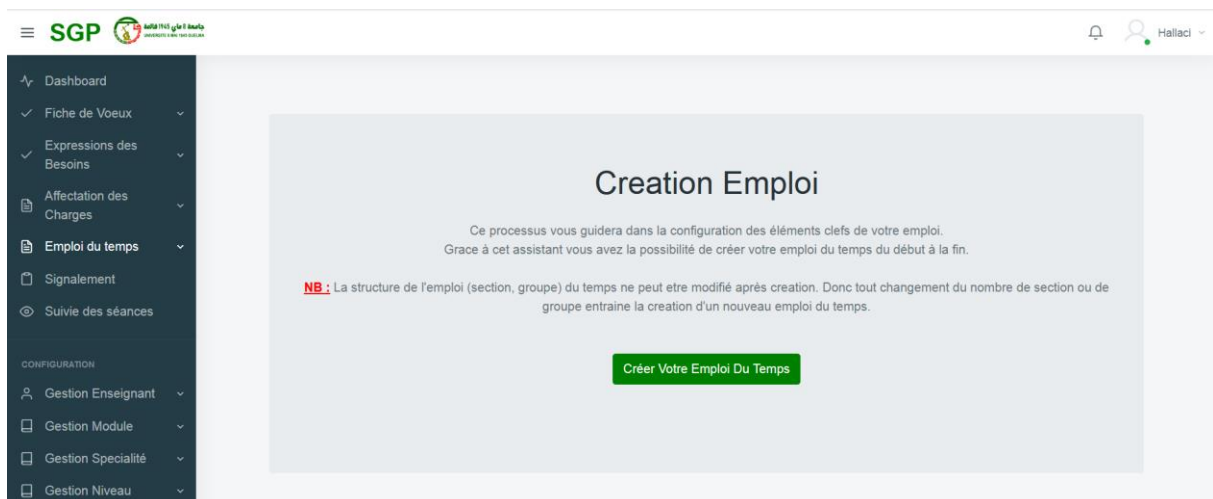


Figure 4.11 : Première étape création emploi du temps

La figure ci-dessous décrit le menu de création de l'emploi du temps. Il permet de créer de créer l'emploi du temps de façon manuelle ou automatique.

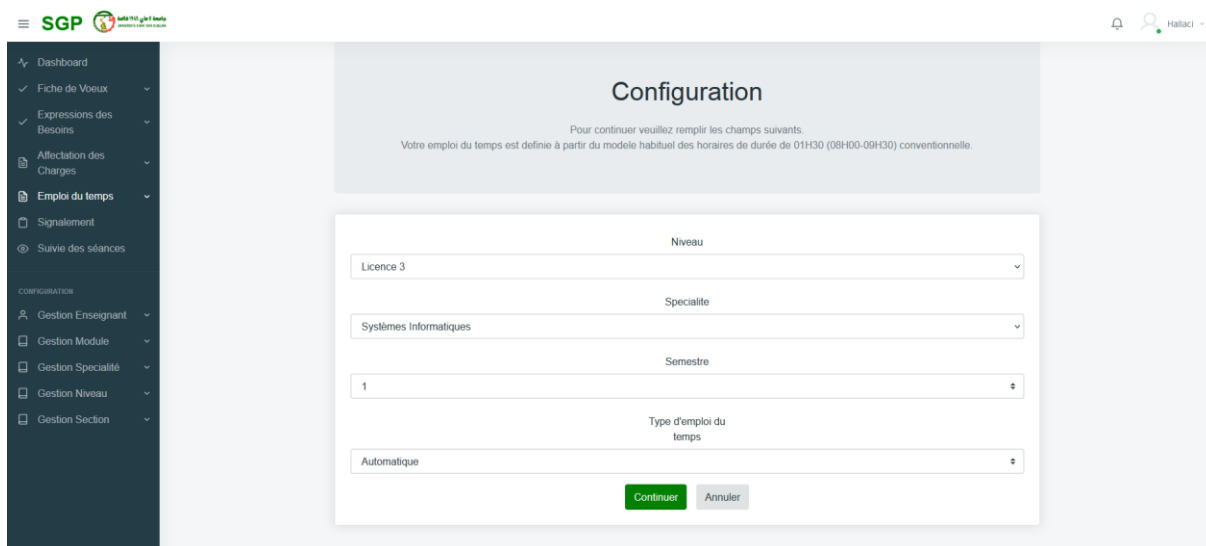


Figure 4.12 : Menu création emploi du temps

Pour éviter les erreurs de duplication, un test est effectué pour prévenir ce type de problème. Le message d’erreur est décrit par la figure ci-dessous.

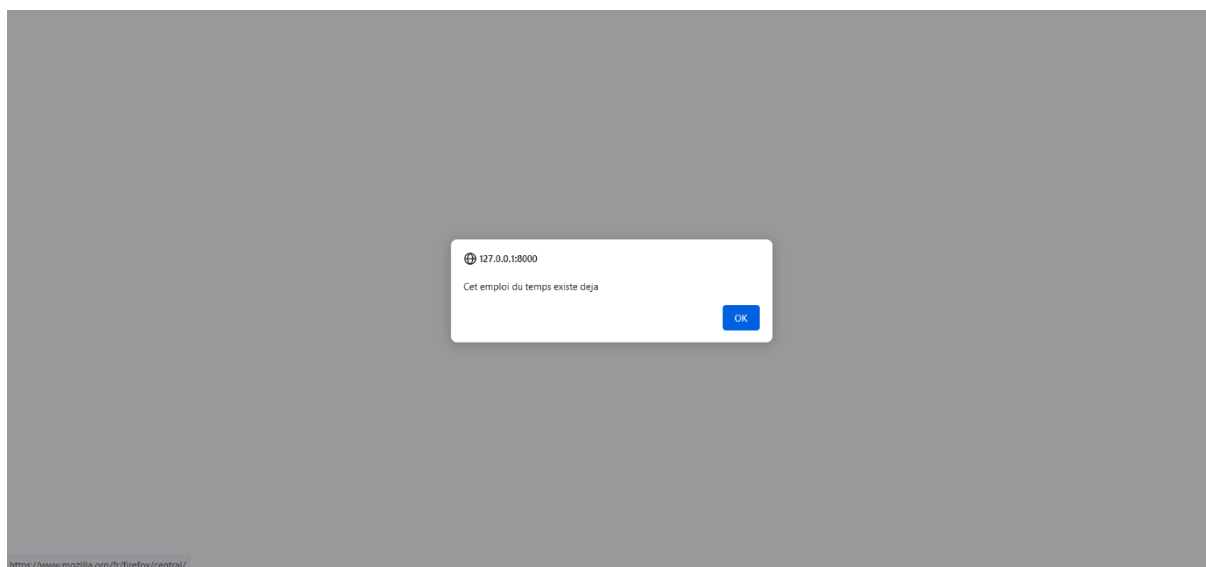


Figure 4.13 : Erreur en cas de duplication

A la fin de la génération l’emploi du temps est affiché au chef de la pédagogie. Notre système n’étant qu’un outil d’aide à la décision, permet de laisser la main pour la modification. Ainsi le chef de la pédagogie peut modifier l’emploi du temps pour ses besoins. La figure ci-dessus décrit la visualisation de l’emploi du temps généré par le système.

Figure 4.14 : Affichage de l'emploi du temps générée

La figure ci-après représente la version imprimée de l'emploi du temps ainsi généré après le système.

		Section 1	
Jour m	Horaire	Groupe 1	Groupe 2
DIMANCHE	08:00-09:00	Interaction Homme Machine Cours (Kouahla M) T2	
	09:00-10:00	Interaction Homme Machine Cours (Kouahla M) T2	
	10:00-11:00	Programmation linéaire TD (Chaoui M) EB.9	
	11:00-12:00	Programmation linéaire TD (Chaoui M) EB.9	
	12:00-13:00		
	13:00-14:00	Système d'exploitation 2 Cours (Seridi A) EB.9	
	14:00-15:00	Système d'exploitation 2 Cours (Seridi A) EB.9	
LUNDI	08:00-09:00	Compilation Cours (Mehnaoui Z) EB.9	
	09:00-10:00	Compilation Cours (Mehnaoui Z) EB.9	
	10:00-11:00	Compilation TD (Mehnaoui Z) T3	
	11:00-12:00	Compilation TD (Mehnaoui Z) T3	
	12:00-13:00		
	13:00-14:00	Probabilités et Statistiques Cours (Ezzebba A) EB.7	
	14:00-15:00	Probabilités et Statistiques Cours (Ezzebba A) EB.7	
MARDI	08:00-09:00	Compilation TP (Benzarek A) EB.12	Système d'exploitation 2 TP (Seridi A) EB.13
	09:00-10:00	Compilation TP (Benzarek A) EB.12	Système d'exploitation 2 TP (Seridi A) EB.13
	10:00-11:00	Système d'exploitation 2 TP (Seridi A) EB.15	Interaction Homme Machine TP (Mabrek Z) EB.13
	11:00-12:00	Système d'exploitation 2 TP (Seridi A) EB.15	Interaction Homme Machine TP (Mabrek Z) EB.13
	12:00-13:00		
	13:00-14:00	Probabilités et Statistiques TD (Ezzebba A) EB.9	
	14:00-15:00	Probabilités et Statistiques TD (Ezzebba A) EB.9	
MERCREDI	08:00-09:00	Interaction Homme Machine TD (Mabrek Z) EB.7	
	09:00-10:00	Interaction Homme Machine TD (Mabrek Z) EB.7	
	10:00-11:00	Génie Logiciel Cours (Djakhdiaakha L) T2	
	11:00-12:00	Génie Logiciel Cours (Djakhdiaakha L) T2	
	12:00-13:00		
	13:00-14:00	Génie Logiciel TD (Djakhdiaakha L) EB.9	
	14:00-15:00	Génie Logiciel TD (Djakhdiaakha L) EB.9	
JEUDI	08:00-09:00	Génie Logiciel TP (Djakhdiaakha L) EB.14	Compilation TP (Benzarek A) EB.13
	09:00-10:00	Génie Logiciel TP (Djakhdiaakha L) EB.14	Compilation TP (Benzarek A) EB.13
	10:00-11:00	Interaction Homme Machine TP (Mabrek Z) EB.12	Génie Logiciel TP (Djakhdiaakha L) EB.11
	11:00-12:00	Interaction Homme Machine TP (Mabrek Z) EB.12	Génie Logiciel TP (Djakhdiaakha L) EB.11
	12:00-13:00		
	13:00-14:00	Economie numérique et veille stratégique TD (Khabiz A) EB.9	
	14:00-15:00	Economie numérique et veille stratégique TD (Khabiz A) EB.9	

Figure 4.15 : Impression de l'emploi du temps

3 Implémentation de l'ontologie :

Les figures suivantes décrivent l'implémentation de nos deux ontologies modulaires.

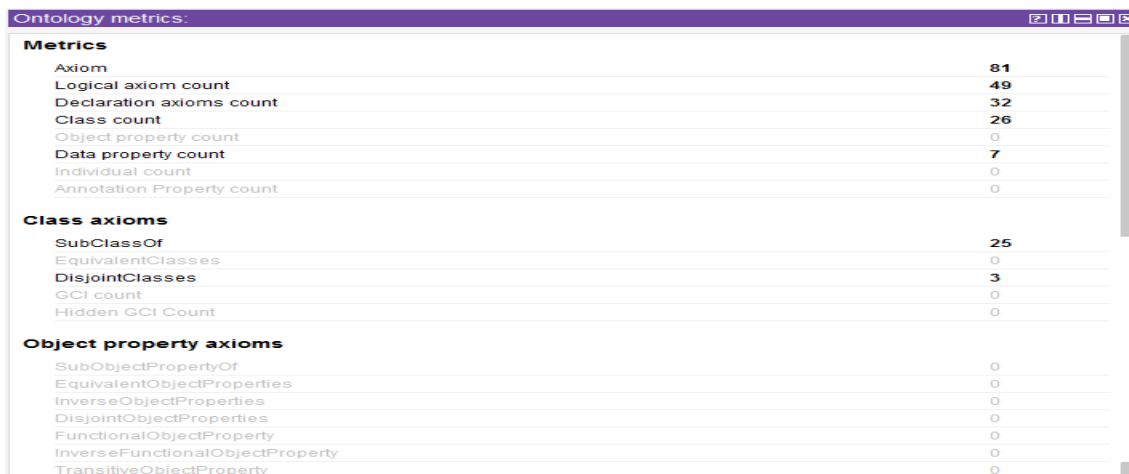
3.1 Module « Acteur » :

La figure suivante décrit la création de l'ontologie avec Owlready2. On y aperçoit la création des différentes classes.

```
Entrée [2]: with onto:
  class Personne(Thing): pass
  class id_personne (Personne >> int, FunctionalProperty):pass
  class nom (Personne >> str, FunctionalProperty):pass
  class prenom (Personne >> str, FunctionalProperty):pass
  class date_naissance (Personne >> str, FunctionalProperty):pass
  class Visiteur(Personne): pass
  class Etudiant(Personne): pass
  class Employe(Personne): pass
  class Enseignant(Employe): pass
  class Staff_administratif(Employe): pass
  class Staff_technique(Employe): pass
  AllDisjoint([Etudiant, Staff_technique,Enseignant])
```

Figure 4.16 : Code de création de l'ontologie Acteur

La figure ci-dessous décrit la visualisation des informations de notre ontologie dans Protégé¹.



Ontology metrics:	
Metrics	
Axiom	81
Logical axiom count	49
Declaration axioms count	32
Class count	26
Object property count	0
Data property count	7
Individual count	0
Annotation Property count	0
Class axioms	
SubClassOf	25
EquivalentClasses	0
DisjointClasses	3
GCI count	0
Hidden GCI Count	0
Object property axioms	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0

Figure 4.17 : Details de l'ontologie Acteur

La figure suivante décrit la visualisation de la hiérarchie des classes.

¹ <https://protege.stanford.edu>

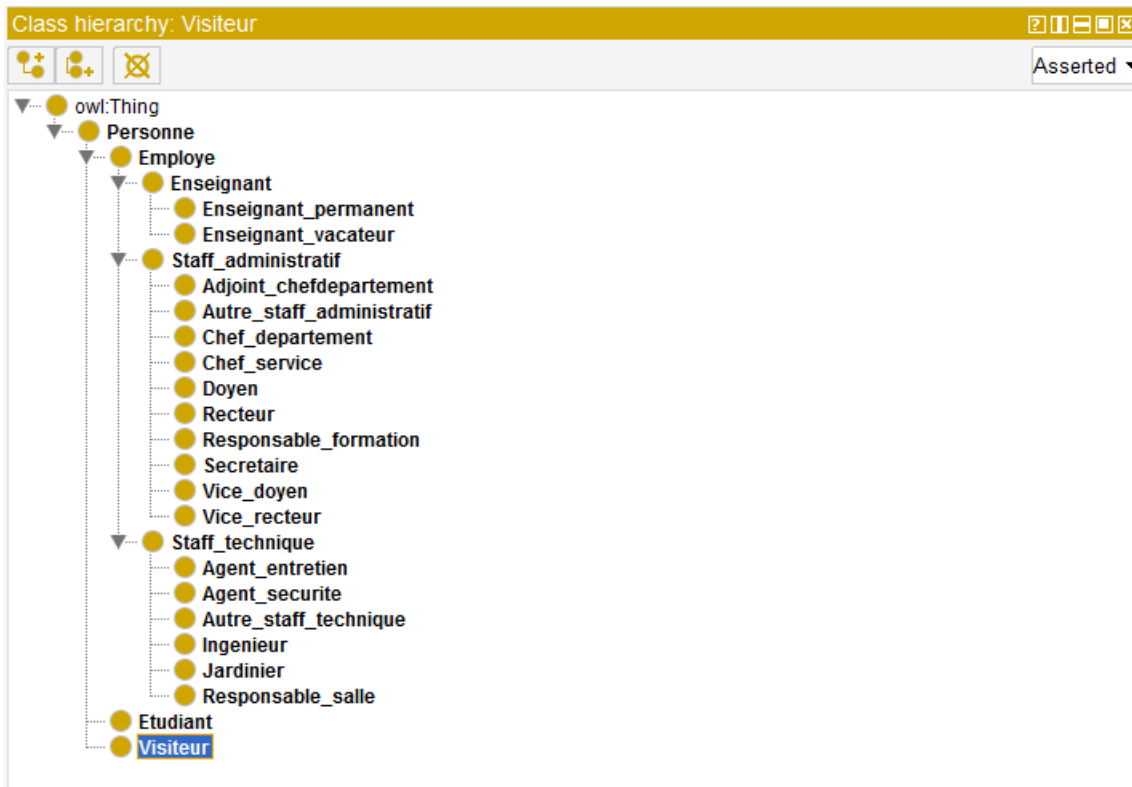


Figure 4.18 : Hiérarchie des classes de l'ontologie Acteur

La figure suivante décrit la visualisation de la hiérarchie des relations. Notre ontologie ne contient aucune relation à part celle d'hérarchie.

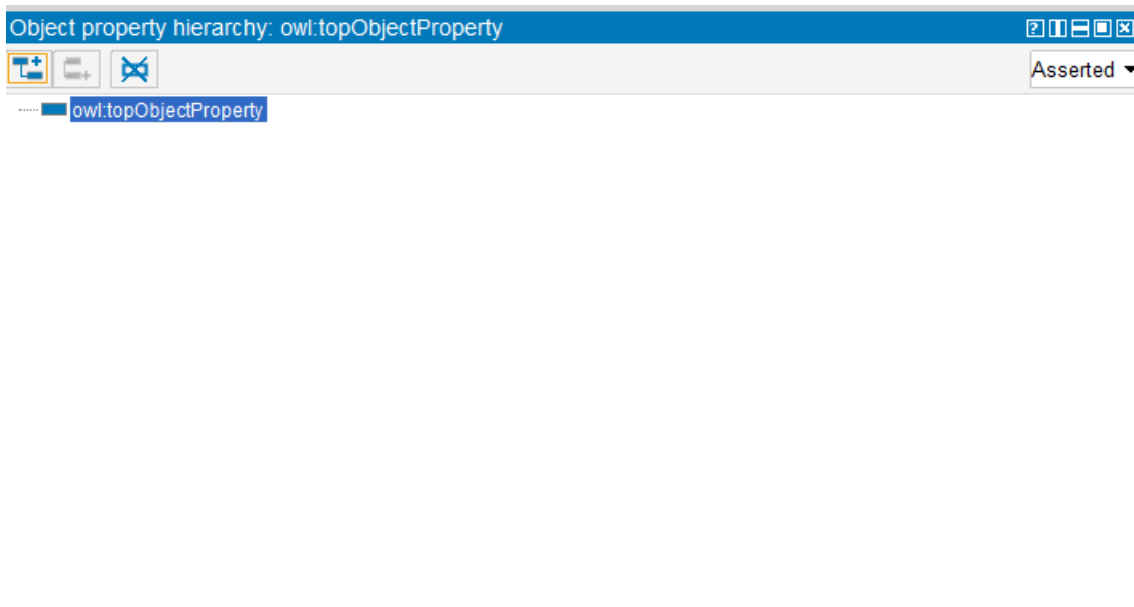


Figure 4.19 : Hiérarchie des relations de l'ontologie Acteur

La figure suivante présente l'ensemble des attributs de notre ontologie.

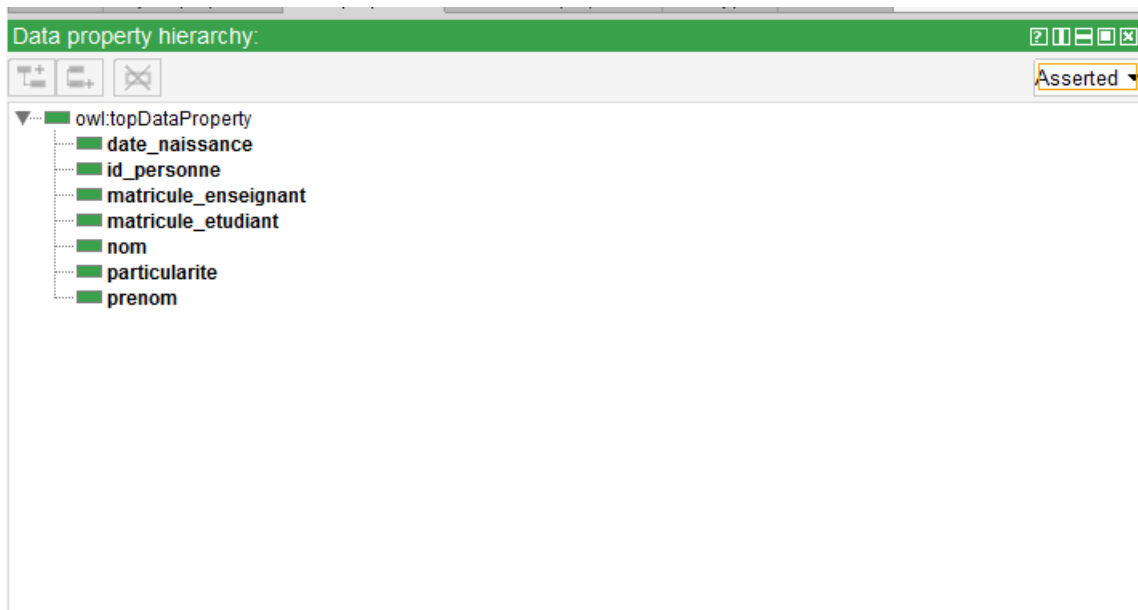


Figure 4.20 : Hiérarchie des attributs de l'ontologie Acteur

3.2 Le module « Classe intelligente » :

La figure suivante décrit la création de l'ontologie avec Owlready2.

```

Entrée [28]: with onto:
class Position(Thing):pass
class Horloge(Thing):pass
class Contexte(Thing):pass

Entrée [29]: with onto:
class Humidite_air(Contexte):pass
class Light(Contexte):pass
class Attendance(Contexte):pass
class Temperature_air(Contexte):pass
class Gaz(Contexte):pass

Entrée [30]: with onto:
class humidite_valeur(Humidite_air >> int, FunctionalProperty):pass
class lux(Light >> int, FunctionalProperty):pass
class nombre_personne(Attendance >> int, FunctionalProperty):pass
class temperature_valeur(Temperature_air >> int, FunctionalProperty):pass
class temperature_ideal(Temperature_air >> int, FunctionalProperty):pass
class volume_gaz(Gaz >> int, FunctionalProperty):pass
class x(Position >> int, FunctionalProperty):pass
class y(Position >> int, FunctionalProperty):pass

Entrée [31]: with onto:
class genere(ObjectProperty, FunctionalProperty):
domain = [Capteur]
range = [Contexte]
class sepasse(ObjectProperty, FunctionalProperty):
domain = [Horloge]
range = [Contexte]
class fonctionneselon(ObjectProperty, FunctionalProperty):
domain = [Piece_activite]
range = [Horloge]
class setrouve(ObjectProperty, FunctionalProperty):
domain = [Piece_activite]

```

Figure 4.21 : Code de création de l'ontologie Classe intelligente

La figure ci-dessous décrit la visualisation des informations de notre ontologie dans Protégé².

Ontology metrics:	
Metrics	
Axiom	467
Logical axiom count	280
Declaration axioms count	187
Class count	143
Object property count	11
Data property count	34
Individual count	0
Annotation Property count	0
Class axioms	
SubClassOf	142
EquivalentClasses	0
DisjointClasses	3
GCI count	0
Hidden GCI Count	0
Object property axioms	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	11
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0

Figure 4.22 : Détails de l'ontologie Classe intelligente

La figure suivante décrit la visualisation de la hiérarchie des classes.

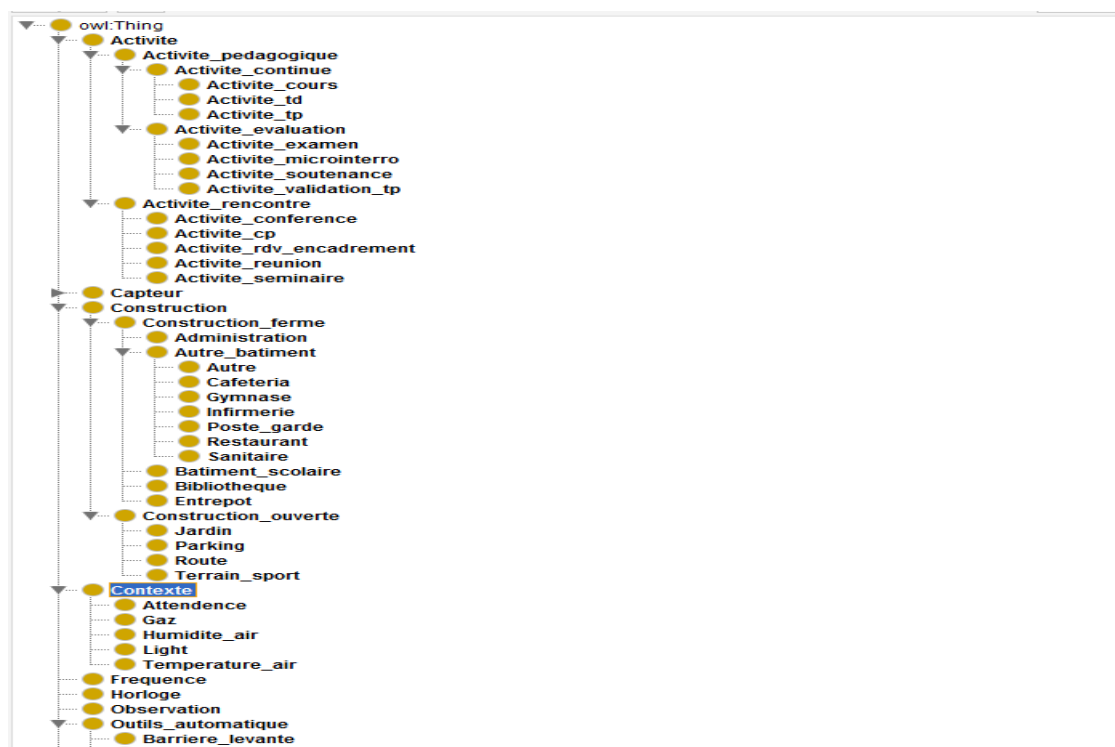


Figure 4.23 : Hiérarchie de classe de l'ontologie Classe intelligente

² <https://protege.stanford.edu>

La figure suivante décrit la visualisation de la hiérarchie des relations.

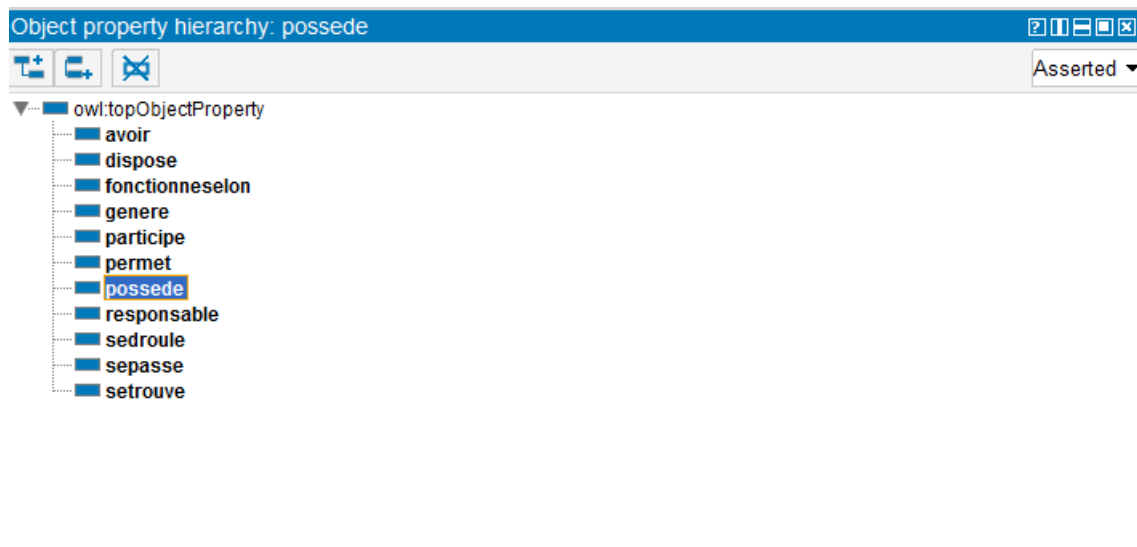


Figure 4.24 : Hiérarchie de relation de l'ontologie Classe Intelligente

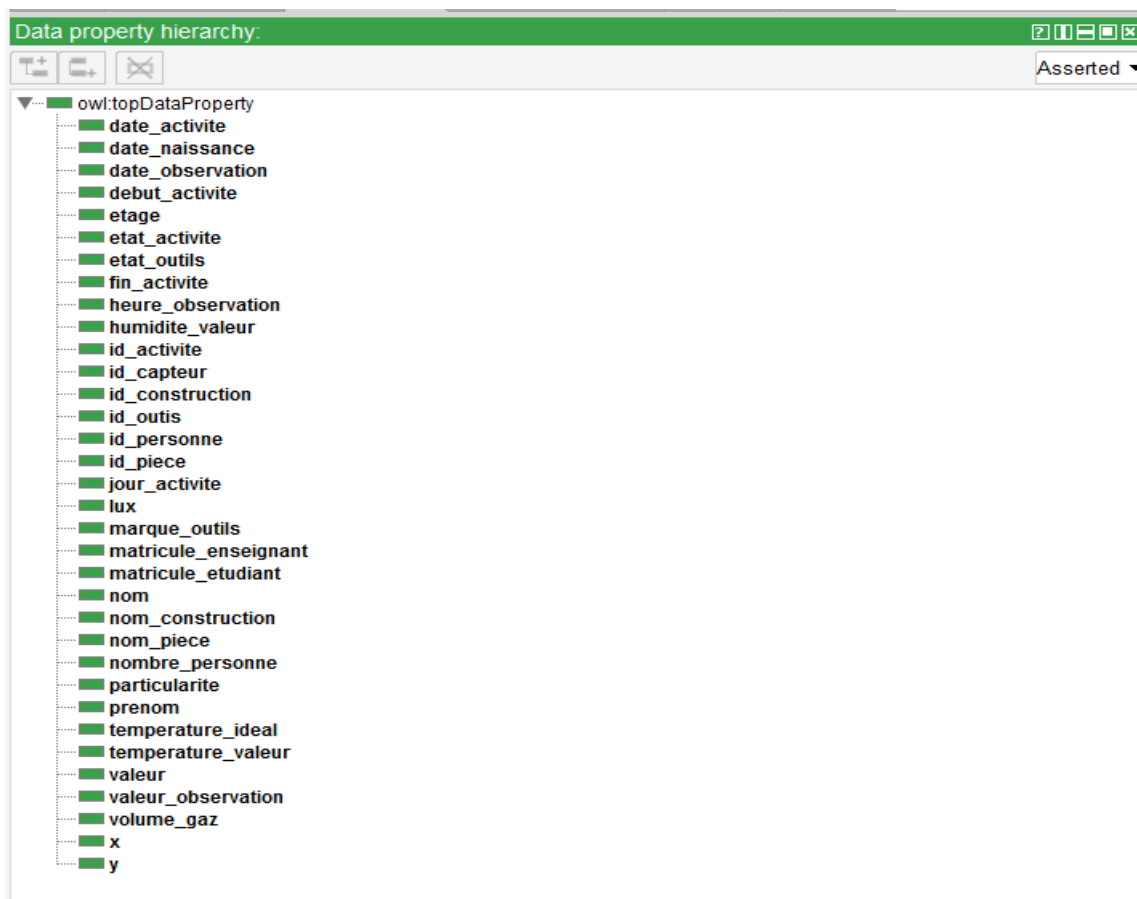


Figure 4.25 : Hiérarchie des attributs de l'ontologie Classe intelligente

4 Architecture du système :

Pour l'implémentation de notre architecture, nous avons utilisé un ensemble de capteurs (voir figures ci-dessous). Parmi ses capteurs on distingue un lecteur de carte intelligent que nous avons développé, plusieurs capteurs ultrasons, un capteur de température, et plusieurs interrupteurs de fin de course. Ces capteurs constitue le niveau physique de notre architecture



Figure 4.26 : Lecteur intelligent de carte développé par nous



Figure 4.27 : Capteur de température

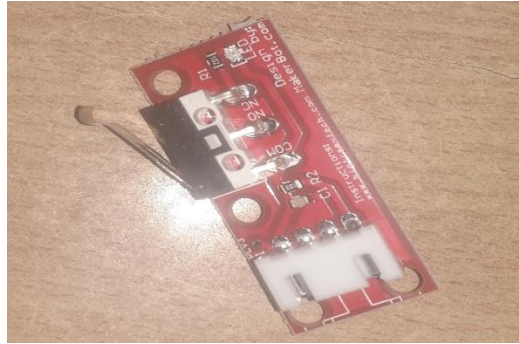


Figure 4.28 : *Interrupteur de fin de course*

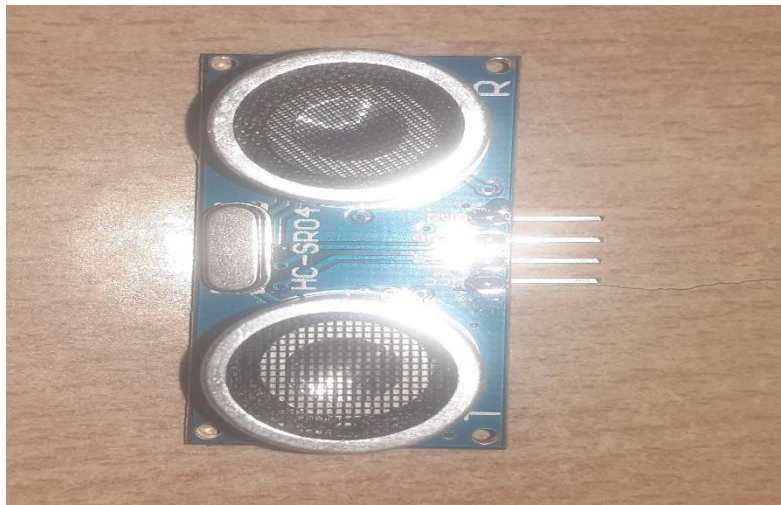


Figure 4.29 : *Capteur ultrason*

Les données des capteurs sont envoyées à l'ontologie stockés dans le niveau Edge à l'aide d'un routeur internet (voir figure ci-dessous). Le routeur n'est pas connecté à internet entre le Fog et le Edge. On effectue un raisonnement par rapport à ces données grâce à l'ontologie et on effectue les actions à faire en temps réel (comme l'ouverture de la porte, etc.). Ensuite ces données sont transférées au niveau Fog. A ce niveau, on stocke les informations indispensables au système de gestion de la pédagogie et autres systèmes de gestion stockés au niveau du Fog. Ensuite l'ensemble des données seront envoyés au cloud pour un stockage permanent. La liaison entre le Cloud et le Fog se fait via internet. Au niveau du Fog on pourra effectuer les analyses de donnée avancer pour afficher les tableaux de bord global. Cependant nous avons pu que faire l'implémentation du niveau Edge et du niveau Fog. L'implémentation du niveau Cloud fera l'objet des prochaines perspectives de notre travail.

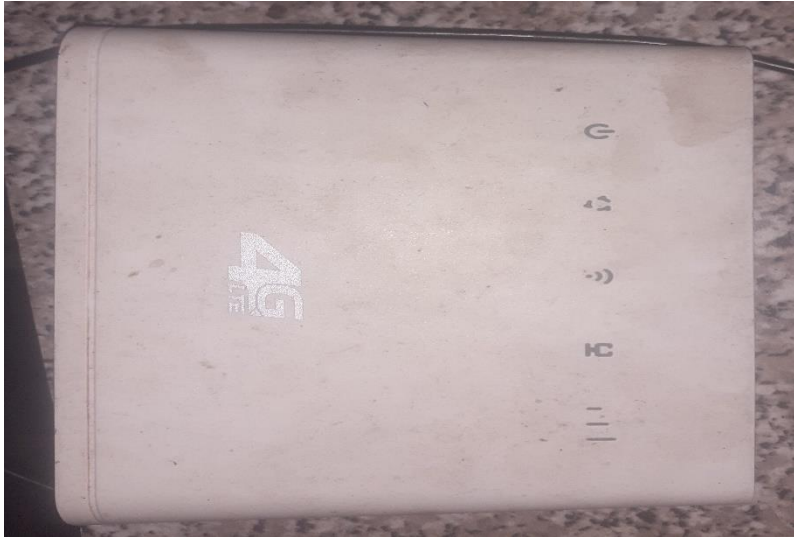


Figure 4.30 : Routeur

Conclusion :

Nous avons dans ce chapitre décrit les outils et les étapes de l'implémentation de la solution proposée. Nous avons utilisé des scénarios pour la mise en exécution de notre solution.

Conclusion Générale

Dans ce mémoire de master, nous avons présenté l'état de l'art sur le campus intelligent. A partir de ces informations, nous avons évoqué les limites des travaux existants dans le domaine. De plus ce travail se trouve être une continuation de notre système de gestion de la pédagogie car la pédagogie est une activité clé au sein d'un campus intelligent. Ainsi nous avons réalisé dans ce travail plusieurs contributions :

- ✓ La génération automatique de l'emploi de temps qui va améliorer le système manuel existant.
- ✓ Une ontologie modulaire pour le stockage et traitement des données issues de campus intelligent.
- ✓ Une architecture Cloud, Fog, Edge pour notre système

Nous avons pu réaliser l'ensemble de nos objectifs à part celle de l'implémentation de la couche Cloud. Ainsi l'implémentation du cloud et celui de notre système complet dans un cas réel feront l'objet de nos futures perspectives.

En attendant nos futures perspectives, ce travail nous a appris à maîtriser les concepts relatifs aux différentes architectures d'applications, à l'IoT, aux ontologies, et à l'usage des algorithmes génétiques dans un problème réel.

Références

- [1] N. Min-Allah and S. Alrashed, “Smart campus—A sketch,” *Sustain. Cities Soc.*, vol. 59, no. December 2019, p. 102231, 2020, doi: 10.1016/j.scs.2020.102231.
- [2] K. U. Sarker and M. Sohail, “Kids ’ Smart Campus Ontology to Retrieve Interest,” *2019 4th MEC Int. Conf. Big Data Smart City*, pp. 1–4, 2019.
- [3] B. Sigl, M. Golub, and V. Mornar, “Solving timetable scheduling problem using genetic algorithms,” *Proc. Int. Conf. Inf. Technol. Interfaces, ITI*, pp. 519–524, 2003, doi: 10.1109/ITI.2003.1225396.
- [4] D. Mittal, H. Doshi, M. Sunasra, and R. Nagpure, “Automatic Timetable Generation using Genetic Algorithm,” *Ijarcce*, vol. 4, no. 2, pp. 245–248, 2015, doi: 10.17148/ijarcce.2015.4254.
- [5] A. R. East, “Timetable Scheduling via Genetic Algorithm,” National University of Ireland, Galway, 2019.
- [6] V. Sapru, K. Reddy, and B. Sivaselvan, “Time table scheduling using Genetic Algorithms employing guided mutation,” *2010 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2010*, pp. 335–339, 2010, doi: 10.1109/ICCIC.2010.5705788.
- [7] A. Laiadi, “Une approche sémantique pour la gestion de données issues de l’Internet des objets dans un contexte de Big Data,” Université 8 Mai 1945 Guelma, 2020.
- [8] D. Boukara, “Utilisation des ontologies pour l’intégration des IoTs dans les processus métiers,” Université 8 Mai 1945 Guelma, 2019.
- [9] F. Gandon, “Ontologies informatiques,” 2006. <https://interstices.info/ontologies-informatiques/> (accessed Jun. 06, 2022).
- [10] S. Despres, “Construction dune ontologie modulaire. Application au domaine de la cuisine numérique,” *Rev. d’Intelligence Artif.*, vol. 30, no. 5, pp. 509–532, 2016, doi: 10.3166/RIA.30.509-532.
- [11] M. El Ghosh, H. Naja, H. Abdulrab, and M. Khalil, “Application of Ontology Modularization for Building a Criminal Domain Ontology,” *Lect. Notes Comput. Sci.*

- (including *Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics*), vol. 10791, pp. 394–409, 2018, doi: 10.1007/978-3-030-00178-0_27.
- [12] S. Sagar, I. Rebai, and M. Khemaja, “Ontologie modulaire pour la fabrication et l’ exploitation de vêtements intelligents dédiés au sport,” 2012.
- [13] Q. Qi and F. Tao, “A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing,” *IEEE Access*, vol. 7, pp. 86769–86777, 2019, doi: 10.1109/ACCESS.2019.2923610.
- [14] D. Basavaraj and S. Tayeb, “Limitations and challenges of fog and edge-based computing,” *2021 IEEE Int. IOT, Electron. Mechatronics Conf. IEMTRONICS 2021 - Proc.*, 2021, doi: 10.1109/IEMTRONICS52119.2021.9422535.
- [15] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016, doi: 10.1109/JIOT.2016.2579198.
- [16] D. Linthicum, “Edge computing vs. fog computing: Definitions and enterprise uses.” [https://www.ewinsonic.com/blog-Edge computing vs fog computing Definitions and enterprise uses.html](https://www.ewinsonic.com/blog-Edge-computing-vs-fog-computing-Definitions-and-enterprise-uses.html) (accessed Jun. 05, 2022).
- [17] A. Yousefpour *et al.*, “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” *J. Syst. Archit.*, vol. 98, no. February, pp. 289–330, 2019, doi: 10.1016/j.sysarc.2019.02.009.
- [18] Javatpoint, “Fog computing vs. Cloud computing.” <https://www.javatpoint.com/fog-computing-vs-cloud-computing#:~:text=The main difference between fog,and can be processed locally.> (accessed Jun. 05, 2022).
- [19] S. D. Nagowah, H. Ben Sta, and B. A. Gobin-Rahimbux, “Towards achieving semantic interoperability in an IoT-enabled smart campus,” in *5th IEEE International Smart Cities Conference, ISC2 2019*, 2019, pp. 593–598, doi: 10.1109/ISC246665.2019.9071694.
- [20] S. D. Nagowah, R. Abou, and R. Bayrouni, “An Ontology for an IoT-enabled Smart Classroom in a University Campus,” *2019 Int. Conf. Comput. Intell. Knowl. Econ.*, pp. 626–631, 2019, doi: 10.1109/ICCIKE47802.2019.9004369.

- [21] B. Hirsch and J. W. P. Ng, "Education beyond the cloud: Anytime-anywhere learning in a smart campus environment," *2011 Int. Conf. Internet Technol. Secur. Trans. ICITST 2011*, no. December, pp. 718–723, 2011.
- [22] M. W. Sari, P. W. Ciptadi, and R. H. Hardyanto, "Study of Smart Campus Development Using Internet of Things Technology," *J. Phys. Conf. Ser.*, vol. 755, no. 1, p. 011001, Oct. 2016, doi: 10.1088/1742-6596/755/1/011001.
- [23] K. Stancin, P. Posic, and D. Jaksic, "Ontologies in education – state of the art," *Educ. Inf. Technol.*, vol. 25, no. 6, pp. 5301–5320, 2020, doi: 10.1007/s10639-020-10226-z.
- [24] B. Goertzel, "Creating Internet Intelligence: Wild Computing, Distributed Digital Consciousness, and the Emerging Global Brain," *IFSR Int. Ser. Syst. Sci. Eng.*, vol. 18, 2002, [Online]. Available: <https://books.google.dz/books?id=Vnzb-xLdvv8C&pg=PA172&lpg=PA172&dq=We+consider+ontologies+to+be+domain+theories+that+specify+a+domain+specific+vocabulary+of+entities,+classes,+properties,+predicates,+and+functions,+and+to+be+a+set+of+relationships+that+n>.
- [25] M. O. Alassafi and M. A. Alqarni, "An Effective Knowledge-Based Modeling Approach towards a ' Smart-School Care Coordination System ' for Children and Young People with Special Educational Needs and Disabilities," 2020.
- [26] Y. Chabot, G. Privat, and F. Ramparany, "Le sens du sens : les ontologies, ce n'est pas (que) de la philosophie !," 2017. <https://hellofuture.orange.com/fr/le-sens-du-sens-les-ontologies-ce-nest-pas-que-de-la-philosophie/> (accessed May 21, 2022).
- [27] D. Boukara, "Utilisation des ontologies pour l'Intégration d'Internet des Objets dans la gestion des processus métier," Université 8 Mai 1945, 2019.
- [28] V. Maran, A. Machado, G. M. Machado, I. Augustin, and J. P. M. De Oliveira, "Domain Content Querying Using Ontology-Based Context-Awareness in Information Systems," *Data Knowl. Eng.*, 2018, doi: 10.1016/j.datak.2018.03.003.
- [29] Futura Tech, "Cloud computing : qu'est-ce que c'est?" <https://www.futura-sciences.com/tech/definitions/informatique-cloud-computing-11573/> (accessed Jun. 11, 2022).

- [30] B. Confais, “Conception d ’ un système de partage de données adapté à un environnement de Fog Computing,” Université de Nantes, 2018.
- [31] IoT Industriel, “Fog Computing.” <https://iotindustriel.com/glossaire-iiot/fog-computing/> (accessed Jun. 11, 2022).
- [32] N. SAKOVICH, “Fog Computing vs. Cloud Computing for IoT Projects,” 2019. <https://www.sam-solutions.com/blog/fog-computing-vs-cloud-computing-for-iot-projects/> (accessed Jun. 06, 2022).
- [33] Red Hat, “Une architecture d’edge computing, qu’est-ce que c’est?,” 2021. <https://www.redhat.com/fr/topics/edge-computing/what-is-edge-architecture#:~:text=Une architecture d%27edge computing est une architecture informatique distribuée,même aux extrémités du réseau.>
- [34] Y. Houam, “Les Algorithmes Génétiques Multiobjectifs,” Université Mohamed Khider - Biskra, 2013.
- [35] I. Chouchani, “Utilisation d’un algorithme génétique pour la composition de services Web,” Université du Quebec à Montréal, 2010.
- [36] Alban, “Algorithme Génétique.” <https://ledatascientist.com/algorithme-genetique/> (accessed Jun. 06, 2022).
- [37] F. Mehidid, “Algorithme Génétique,” UNIVERSITÉ ABDELHAMID IBN BADIS-MOSTAGANEM, 2013.
- [38] G. Ashour, A. Al-dubai, and I. Romdhani, “Ontology-based Course Teacher Assignment within Universities,” vol. 11, no. 7, pp. 720–728, 2020.
- [39] N. Verstaevel, G. Garzone, T. Monteil, N. Guermouche, J. Barthelemy, and P. Perez, “An ontology based context-aware architecture for smart campus applications,” 2018 *IEEE Intl Conf Parallel Distrib. Process. with Appl. Ubiquitous Comput. Commun. Big Data Cloud Comput. Soc. Comput. Networking, Sustain. Comput. Commun.*, pp. 1056–1063, 2018, doi: 10.1109/BDCloud.2018.00158.
- [40] S. D. Nagowah, “An Ontology for an IoT-enabled Smart Parking in a University Campus,” no. July 2021, 2019, doi: 10.1109/ISC246665.2019.9071751.

- [41] S. D. Nagowah, “An Ontology for an IoT-enabled Smart Classroom in a University Campus,” no. March 2020, 2019, doi: 10.1109/ICCIKE47802.2019.9004369.
- [42] R. Fleiner, “A Case Study on Linked Data for University Courses,” vol. 3, pp. 265–276, 2017, doi: 10.1007/978-3-319-55961-2.
- [43] Z. Samia, R. Khaled, and Z. Warda, “Multi-Agent Systems and Ontology for Supporting Management System in Smart School,” *2018 3rd Int. Conf. Pattern Anal. Intell. Syst.*, pp. 1–8.
- [44] H. Chung and J. Kim, “An Ontological Approach for Semantic Modeling of Curriculum and Syllabus in Higher Education,” vol. 6, no. 5, pp. 365–369, 2016, doi: 10.7763/IJNET.2016.V6.715.
- [45] S. Missaoui, “Towards an Ontology for Representing a Student ’ s Profile in Adaptive Gamified Learning System,” vol. 2, no. Ic3k, pp. 149–156, 2020, doi: 10.5220/0010107301490156.
- [46] M. Majid, M. F. Hayat, F. Z. Khan, and M. Ahmad, “Ontology-Based System for Educational Program Counseling,” 2021, doi: 10.32604/iasc.2021.017840.
- [47] J. Chen, “Developing Educational Ontology : A Case Study in Physics,” no. 3663, 2018.
- [48] S. Somsuphprungyos, M. Buranaruch, S. Boonbrahm, N. S. Thammarat, and N. Electronics, “A Framework of Decision Support System based on Integrated Data for Electricity Management in Campus,” vol. 7, no. 2, pp. 183–193, 2017.
- [49] C. Obeid *et al.*, “Ontology-based recommender system in higher education,” 2019.
- [50] X. Jing, S. Li, J. Cheng, and J. Guo, “Multidimensional situational information fusion method for energy saving on campus,” vol. 38, pp. 4793–4807, 2020, doi: 10.3233/JIFS-191513.
- [51] T. Nguyen-anh, “An IoT Reconfiguration Framework Applied Ontology-based Modeling and Bayesian-based Reasoning for Context Management,” pp. 540–545, 2019.
- [52] N. A. Alrehaili *et al.*, “Ontology-Based Smart System to Automate Higher Education Activities,” vol. 2021, 2021.

- [53] S. Somsuphprungyos, S. Boonbrahm, and M. Buranarach, “Inferring Students’ Activity Using RFID and Ontology.”
- [54] S. Somsuphprungyos and S. Boonbrahm, “A Recommender of Transportation Planning in Campus Using Ontology,” vol. 1.
- [55] Y. Tsukagoshi, S. Egami, Y. Sei, Y. Tahara, and A. Ohsuga, “Ontology-Based Correlation Detection Among Heterogeneous Data Sets : A Case Study of University Campus Issues,” pp. 33–40, 2020, doi: 10.1109/AIKE48582.2020.00014.
- [56] “Algorithme génétique.” http://igm.univ-mlv.fr/~dr/XPOSE2013/tleroux_genetic_algorithm/fonctionnement.html (accessed Jun. 06, 2022).
- [57] M. Abbaszadeh, S. Saeedvand, and H. Asbahi Mayani, “Solving University Scheduling Problem with a Memetic Algorithm,” *IAES Int. J. Artif. Intell.*, vol. 1, no. 2, pp. 79–90, 2012, doi: 10.11591/ij-ai.v1i2.512.
- [58] A. Lima, “Sélection de tournoi (AG).” <https://fr.acervolima.com/selection-de-tournoi-ag/> (accessed Jun. 06, 2022).
- [59] Khayyam90, “Les algorithmes génétiques,” 2005. <https://khayyam.developpez.com/articles/algo/genetic/#:~:text=La sélection par tournoi▲,au hasard dans la population.> (accessed Jun. 06, 2022).
- [60] M. Keet, “Methodologies for Ontology Development,” 2020. [https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_and_Computation_Fundamentals/Book%3A_An_Introduction_to_Ontology_Engineering_\(Keet\)/06%3A_Methods_and_Methodologies/6.01%3A_Methodologies_for_Ontology_Development](https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_and_Computation_Fundamentals/Book%3A_An_Introduction_to_Ontology_Engineering_(Keet)/06%3A_Methods_and_Methodologies/6.01%3A_Methodologies_for_Ontology_Development) (accessed Jun. 06, 2022).
- [61] C. Roussey and J.-P. Chanet, “Le premier module d’une ontologie agricole sur la protection des cultures : Agronomic Taxon,” 2013.
- [62] S. Despres, “Construction d’une ontologie modulaire pour l’univers de la cuisine numérique Sylvie Despres To cite this version : HAL Id : hal-01010222,” 2014.
- [63] S. Cardoso *et al.*, “De l’intérêt des ontologies modulaires . Application à la modélisation

de la prise en charge de la SLA To cite this version : HAL Id : hal-01839571 De l' intérêt des ontologies modulaires . Application à la modélisation de la prise en charge de la SLA,” 2018.

- [64] C. Garcia-Montero, “MQTT : comment fonctionne ce protocole ?,” 2020, [Online]. Available: <https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1440686-mqtt-comment-fonctionne-ce-protocole/>.
- [65] P. Chopinet, “Python : Comment faire une api web avec Flask,” 2021. <https://blog.jaaj.dev/2021/04/20/Comment-faire-une-api-web-en-python.html> (accessed Jun. 11, 2022).
- [66] Jiba, “Owlready : un module Python pour manipuler les ontologies OWL,” 2017. <https://linuxfr.org/news/owlready-un-module-python-pour-manipuler-les-ontologies-owl> (accessed Jun. 06, 2022).