

UNIVERSITÉ DE YAOUNDÉ I

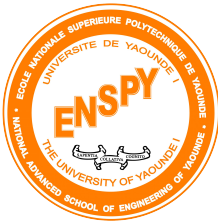
ÉCOLE NATIONALE SUPÉRIEURE
POLYTECHNIQUE

DÉPARTEMENT DE GÉNIE
INFORMATIQUE

UNIVERSITY OF YAOUNDE I

NATIONAL ADVANCED SCHOOL OF
ENGINEERING

DEPARTMENT OF COMPUTER
ENGINEERING



THÈME :

Conception d'un Système Tutoriel Intelligent d'aide à la pose de Diagnostic Médical

Par

CHIMI YOUKAP ORÉAL
KENFACK CYRILLE
NDJEBAYI DAMARIS
NGUEN KÉVINA
TCHEUJO TETA DERRICK

Sous la supervision du :

DR BERNABÉ BATCHAKUI

ANNÉE ACADÉMIQUE 2021-2022

RÉSUMÉ

La formation médicale se tourne principalement vers le paradigme de l'apprentissage. Toutefois, il y a une charge énorme de travail dans l'alternance entre les heures de cours et les séances pratiques à l'hôpital, ce qui ne promet pas toujours que des notions fondamentales comme celle sur la pose d'un diagnostic ont été assimilées correctement par nos médecins. Afin de proposer une formation de qualité, la simulation, le e-learning et les patients virtuels qui sont des simulations informatiques interactives de scénarios cliniques, permettant l'apprentissage, la pratique et l'évaluation du raisonnement clinique, de la recherche d'information, de la prise de décision en contexte d'incertitude et même du travail en équipe sont de plus en plus utilisés dans des écoles de médecine, et même sur le terrain, dans de nombreux hôpitaux. Dans le présent document, nous décrivons une approche de conception, de modélisation et d'implémentation d'un système tutoriel intelligent d'aide à la pose de diagnostic médical en nous basant sur un patient virtuel linéaire.

Mots-clés : Apprentissage, Système tutoriel Intelligent, Patient virtuel, médecine

ABSTRACT

Medical education primarily revolves around the learning paradigm. However, there is an enormous workload in the alternation between class hours and practical sessions in the hospital, which does not always promise that fundamental notions such as that on the establishment of a diagnosis have been assimilated. correctly by our doctors. In order to provide quality training, simulation, e-learning and virtual patients which are interactive computer simulations of clinical scenarios, allowing the learning, practice and evaluation of clinical reasoning, information retrieval , decision-making under uncertainty and even teamwork are increasingly used in medical schools, and even in the field, in many hospitals. In this document, we describe an approach for the design, modeling and implementation of an intelligent tutorial system to help with medical diagnosis based on a linear virtual patient.

Keywords : Learning, Intelligent tutorial system, virtual patient, medicine

ABRÉVIATIONS

Acronymes	Significations
STI	Système Tutoriel Intelligent
NLP	Natural Language Processing
CAO	Conception Assisté par Ordinateur
MST	Maladie Sexuellement Transmissible
UMLS	Unified Medical Language System
SNOMED	Systematized Nomenclature of Medicine
ADM	Aide au Diagnostic Médical
VIPS	Virtual Internet Patient Simulation
LIM	Laboratoire d'Informatique Médicale
SGBD	Système de Gestion de Base de Données
3D	Dimension 3
OMS	Organisation Mondiale de la Santé
NLU	Natural Language Understanding
NLG	Natural Language Generation

TABLE DES MATIÈRES

	Page No
Résumé	i
Abstract	ii
Abréviations	iii
Liste des tableaux	viii
Table des figures	ix
INTRODUCTION GÉNÉRALE	x
1 CONCEPTS THEORIQUE ET ÉTAT DE L'ART	1
1.1 Généralités	2
1.1.1 Diagnostic médical	2
1.1.1.1 Définition	2
1.1.1.2 Typologie de diagnostic médical	2
1.1.1.3 Processus pour la pose de diagnostic	2
1.1.1.4 Les étapes du diagnostic	3
1.2 Système Tutoriel Intelligent (STI)	3
1.2.1 Description	3
1.2.2 Architecture d'un STI	4
1.2.2.1 Interface	4
1.2.2.2 Module expert / Modèle du domaine	5
1.2.2.3 Module de l'apprenant	5
1.2.2.4 Module du tuteur	6
1.3 Outils de modélisation des STI	6
1.3.1 Les ontologies	6

1.3.1.1	Définition	6
1.3.1.2	Représentation des ontologies	7
1.3.1.3	Types d'ontologies	7
1.3.1.4	Méthodologie de construction des ontologies	8
1.3.1.5	Les ontologies dans le domaine médical	8
1.3.2	Les réseaux bayésien	9
1.3.2.1	Définition	9
1.3.2.2	Procédé de construction d'un graphe bayésien	9
1.3.2.3	Inférence bayésienne	9
1.3.2.4	Applications	9
1.3.3	Natural Language Processing(NLP)	10
1.3.3.1	Description	10
1.3.3.2	Natural Language Understanding(NLU)	10
1.3.3.3	OntoNLQA	11
1.3.3.4	Natural Language Generation(NLG)	11
1.3.4	Généralité sur la 3D	11
1.3.4.1	Définition	11
1.3.4.2	Généralités sur les logiciels de modélisation tridimensionnelle	12
1.3.4.3	Quelques logiciels de modélisation tridimensionnelle	12
1.4	Quelques SADM	12
1.4.1	ADM (Aide au Diagnostic Médical)	13
1.4.1.1	Contexte de réalisation	13
1.4.1.2	Les points de vue adoptés	13
1.4.1.3	Les limites de ADM	13
1.4.2	VIPS (Virtual Internet Patient Simulation)	13
1.4.2.1	Contexte de réalisation	14
1.4.2.2	Les objectifs de VIPS	14
1.4.2.3	Les limites de VIPS	14
1.4.3	SYDIME	14
1.4.3.1	Avantages	15
1.4.3.2	Limites	15
2	MÉTHODOLOGIE	16
2.1	Analyse	18
2.1.1	Définition de quelques concepts	18

2.1.2	Exigences du système	18
2.1.3	Diagrammes d'analyse	19
2.1.3.1	Diagramme de contexte	19
2.1.3.2	Diagramme de cas d'utilisation	19
2.1.3.3	Diagramme de classe métier	21
2.1.3.4	Activité	22
2.2	Conception	24
2.2.1	Diagramme de séquence technique de l'évaluation d'un apprenant . .	25
2.3	Modélisation	26
2.3.1	Interface	26
2.3.2	Modèle de l'apprenant	28
2.3.2.1	Modèle cognitif	28
2.3.2.2	Modèle affectif	31
2.3.2.3	Modèle inférentiel	32
2.3.3	Modèle du tuteur	33
2.3.3.1	Représentation des connaissances	33
2.3.3.2	Gestion et calculs dans un graphe probabiliste avec la biblio- thèque pyAgrum : Cas du graphe Bayésien	34
2.3.3.3	Communication entre le patient et l'apprenant	35
2.3.3.4	Stratégie pédagogique utilisée	35
2.3.3.5	Critères d'évaluation (dans un exercice sommatif)	36
2.3.3.6	Définition du niveau de l'étudiant	36
2.3.4	Module expert et décision	37
2.3.4.1	Modèle de représentation des connaissances	38
2.3.4.2	Manipulation des connaissances	39
2.4	Architecture du système	39
2.4.1	Architecture globale	39
2.4.2	Architecture détaillée	41
2.5	Implémentation	41
2.5.1	Choix des outils	41
2.5.1.1	Blender	41
2.5.1.2	ReactJS	42
2.5.1.3	DjangoREST Framework (DRF)	43
2.5.1.4	UMLS	44
2.5.1.5	OntoDB(2) (Ontologic DataBase	45

2.5.1.6	PostgreSQL	45
2.5.1.7	OntoNLQA	46
2.6	Déploiement	46
3	EXPÉRIMENTATIONS ET RÉSULTATS	48
3.1	Rappel du problème	49
3.2	Le module de l'expert	49
3.3	Le module du tuteur	49
3.4	Le module de l'apprenant	49
4	DISCUSSION	51
4.1	Avantages	52
4.2	Limites	52
	CONCLUSION GÉNÉRALE	53
	Bibliographie	55

LISTE DES TABLEAUX

TABLE	Page No
2.1 Cas d'utilisation Global	21
2.2 Choix des outils d'implémentation	46

TABLE DES FIGURES

FIGURE	Page No
1.1 Architecture de base d'un STI	4
2.1 Diagramme de contexte	19
2.2 Diagramme de cas d'utilisation	20
2.3 Diagramme de classe métier	22
2.4 Diagramme d'activité du processus d'évaluation formatif	23
2.5 Diagramme d'activité du processus d'évaluation sommatif	24
2.6 Diagramme de séquence technique de l'évaluation d'un apprenant	25
2.7 Interface avec différentes options	26
2.8 Affichage des composantes de test	27
2.9 Simulation de la consultation physique	27
2.10 Discussion entre l'apprenant et le patient virtuel	28
2.11 Type Overlay	28
2.12 Structure globale de la connaissance de l'apprenant	29
2.13 Organigramme de création du réseau Bayésien	31
2.14 Exemple de connaissances	31
2.15 Recette émotionnelle	32
2.16 Architecture Globale	40
2.17 Architecture détaillée de la couche modules	41
2.18 Logo Blender	42
2.19 Logo ReactJS	43
2.20 Logo Django REST	43
2.21 Logo PyTorch	44
2.22 Logo PyAgrum	44
2.23 Aperçu de la terminologie "Maladie" décrite par UMLS	45
2.24 Logo PostgreSQL	46
2.25 Diagramme de déploiement	47

INTRODUCTION GÉNÉRALE

CONTEXTE

En septembre 2019, l'Organisation mondiale de la santé (OMS) instaurait la première journée mondiale de la sécurité des patients pour sensibiliser aux effets indésirables dus à des soins à risque. Dans un rapport daté du troisième trimestre de la même année, l'Organisation soulevait le fait que chaque année, on compte environ 2,6 millions de décès dus à des erreurs médicales. Cela en fait la troisième cause de mortalité après les cancers et les maladies cardiovasculaires. Qu'il s'agisse des erreurs de diagnostic, de prescription ou lors d'opérations chirurgicales ; les conséquences sur les vies des patients sont titanesques. Au regard de ces chiffres et des dommages occasionnés, la sécurité des patients doit être un objectif ancré au cœur des pratiques des professionnels et du système de santé. Il s'agit de prévenir et de réduire les risques, les erreurs et les préjudices causés aux patients dans le cadre de soins de santé. Au cours des dernières années, dans plusieurs pays, la sécurité des patients et les erreurs liées aux soins de santé sont devenues d'importants sujets de préoccupation[5]. Nous pouvons noter que ces accidents arrivent dans la plus part des cas lors des diagnostics fait par les nouveaux médecins (étudiants), car c'est généralement les premiers cas effectifs de maladies traitées par ces derniers. Ces nouveaux médecins n'ont donc pas l'occasion d'expérimenter leurs connaissances avant le stage.

Il est donc intéressant de mettre sur pied un système de formation très efficace afin de transférer les compétences opérationnelles des médecins expérimentés (experts) aux nouvelles générations (apprenants et médecins non expérimentés) d'une part, et d'autre part de permettre à ces derniers de s'entraîner autant que faire se peut avant leur stage clinique. De plus, les étudiants doivent être dans un environnement d'apprentissage qui se rapproche de la vie réelle. Mais la plus part des outils existant n'offrent pas le patient 3D dont ils ont besoin. C'est pour répondre à ces attentes que nous avons fait ce rapport.

Il est question pour nous de faire une modélisation d'un système tutotiel intelligent permettant de former la nouvelle génération à poser de bon diagnostic. Pour cela nous nous sommes pencher sur les concepts qui entrent dans le champs lexical de malade, nous permettant de construire notre ontologie du domaine afin de réaliser la base de connaissance

qui sera utilisée dans la suite pour décrire des cas de maladies et pour construire un patient virtuel 3D représentant le malade.

PROBLÉMATIQUE

Au vu de l'urgence face à l'amélioration des performances de la nouvelle génération dans la pose de diagnostic, nous nous demandons comment à l'aide des systèmes tutoriel intelligent réduire le risque d'erreurs médicales dans la pose de diagnostics ?

OBJECTIFS

L'objectif principal de ce travail est de faire la modélisation d'un système devant :

- Apporter des solutions aux problèmes précédemment cités ;
- Réduire le nombre cas de faux diagnostics ;
- Apporter une assistance professionnelle aux jeunes médecins.

MÉTHODOLOGIE

Pour pouvoir atteindre les objectifs fixés, nous nous sommes appuyés sur plusieurs théories et notions scientifiques parmi lesquelles : la théorie des graphes, les notions de simulations d'émotions, les approches d'intelligence artificielle à base de connaissances, les systèmes multi-agents ainsi que la théorie des probabilités.

PLAN DE TRAVAIL

La suite de ce rapport est structuré comme suite :

- **Concepts théoriques et état de l'art** : où nous commençons par présenter les généralités sur la pose de diagnostic, ensuite nous faisons un rappel sur les systèmes tutoriel intelligent et enfin nous donnons quelques exemples de systèmes existant.
- **Méthodologie** : ici, nous allons présenter notre contribution dans la résolution du problème. Nous présentons une analyse, une conception, une modélisation, une architecture du système, une implémentation et enfin un déploiement de la solution
- **Expérimentation et résultats** : ici, nous allons faire des hypothèses et montrer comment les vérifier.

Nous terminons par une conclusion générale, qui présente le bilan de ce travail et les perspectives.

CONCEPTS THEORIQUE ET ÉTAT DE L'ART

Dans ce chapitre nous présentons les généralités sur la pose de diagnostic, ensuite nous faisons un rappel sur les systèmes tutoriel intelligent et enfin nous donnons quelques exemples de systèmes existant.

Sommaire

1.1	Généralités	2
1.1.1	Diagnostic médical	2
1.2	Système Tutoriel Intelligent (STI)	3
1.2.1	Description	3
1.2.2	Architecture d'un STI	4
1.3	Outils de modélisation des STI	6
1.3.1	Les ontologies	6
1.3.2	Les réseaux bayésien	9
1.3.3	Natural Language Processing(NLP)	10
1.3.4	Généralité sur la 3D	11
1.4	Quelques SADM	12
1.4.1	ADM (Aide au Diagnostic Médical)	13
1.4.2	VIPS (Virtual Internet Patient Simulation	13
1.4.3	SYDIME	14

1.1 Généralités

1.1.1 Diagnostic médical

1.1.1.1 Définition

Le **diagnostic** est la démarche par laquelle le service médical détermine l'affection dont souffre le patient et qui permet de proposer un traitement. Il repose sur la recherche des causes (étiologies) et des effets (symptômes) de l'affection.

1.1.1.2 Typologie de diagnostic médical

Il existe plusieurs types de diagnostic médical :

- **positif** : permet de dire de quoi le malade souffre réellement ;
- **différentiel** : permet de dire toutes les affections qui ressemblent au diagnostic positif mais qui peuvent aussi induire en erreur ;
- **étiologique** : consiste à identifier les causes de la maladie évoquée dans le diagnostic positif

1.1.1.3 Processus pour la pose de diagnostic

- **L'interrogatoire avec le médecin (l'anamnèse)** : ici le médecin est censé interroger le patient en face de lui dans un ordre précis, du plus large au plus précis ;
- **L'examen physique** : qui se déroule en 4 étapes :
 - * L'inspection
 - * La palpation
 - * La percussion
 - * L'auscultation
- **Les examens complémentaires** : si c'est nécessaire, le médecin peut réaliser des examens complémentaires mais cela n'est pas systématique. Ils ont pour but de répondre à une hypothèse diagnostique. Chaque examen correspond à une hypothèse, c'est-à-dire à une question que se pose le médecin.

1.1.1.4 Les étapes du diagnostic

L'établissement d'un diagnostic est une tâche qui s'avère souvent difficile qui comprend les étapes suivantes :

- Déterminer les antécédents des symptômes et recueillir les données pertinentes.
- Effectuer un examen physique.
- Établir un diagnostic provisoire et un diagnostic différentiel.
- Procéder à des investigations (prescrire, revoir et donner suite aux résultats).
- Établir un diagnostic définitif.
- Consulter (un autre médecin pour obtenir des éclaircissements au besoin).
- Fournir des instructions au moment du congé, et assurer une surveillance et un suivi.
- Documenter ces étapes et le raisonnement derrière les décisions prises.

1.2 Système Tutoriel Intelligent (STI)

1.2.1 Description

Les systèmes tutoriels intelligents (STI) sont des environnements d'apprentissage informatisés qui visent à imiter le comportement d'un tuteur humain dans ses capacités d'expert pédagogue et d'expert du domaine. Ainsi, tout comme un tuteur, les logiciels de ce type ont le potentiel d'amener l'apprenant à réaliser une tâche et de fournir des rétroactions pertinentes sur leurs actions.

Les STI sont essentiellement des environnements de résolution de problèmes ou d'exercices. Ils favorisent l'apprentissage dans un domaine précis en guidant et en assistant l'apprenant. Parfois, ils exposent d'abord le contenu du domaine à l'apprenant ; parfois, ils présentent directement les exercices qui permettront d'assimiler les connaissances.

Ces systèmes ont l'intention de prendre en considération l'état de l'apprenant en adaptant les décisions pédagogiques aux besoins particuliers de chaque apprenant. L'adaptabilité se révèle principalement par trois manières différentes :

- présentation dynamique des objectifs d'apprentissage relativement à l'état des connaissances de l'apprenant ;
- ajustement des stratégies ou des tactiques pédagogiques dans un contexte d'assistance en fonction de l'état de l'apprenant ;

- configuration de l'interface ou des rétroactions transmises à l'apprenant en fonction de ses préférences ou de ses besoins.

1.2.2 Architecture d'un STI

L'architecture de base d'un STI est le suivant :

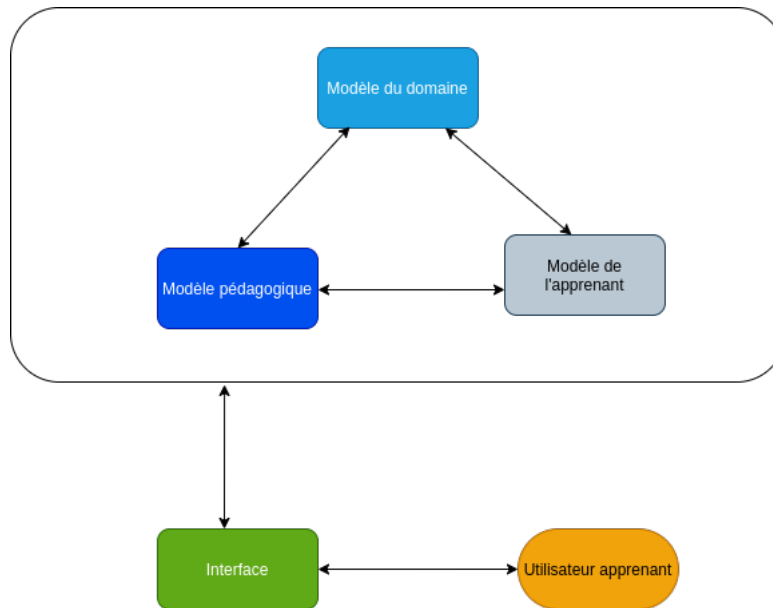


FIGURE 1.1. Architecture de base d'un STI

1.2.2.1 Interface

C'est la couche de communication entre l'apprenant et le STI. l'interface privilégie une approche de conception qui n'obstrue pas l'apprentissage. Pour se faire l'IU pourra nécessiter d'embarquer une intelligence. En outre, ce module devra permettre à l'apprenant de mieux s'exprimer afin de capter ses intentions. Elle varie selon le type d'apprentissage.

Nous distinguons 3 types d'interfaces :

- **Micromonde** : ce type d'interface permet à l'apprenant de construire, simuler et corriger ses propres modèles du monde réel.
- **Simulation** : ce type d'interface permet de faire une représentation des composants d'un domaine. Elle est importante lorsqu'on veut permettre à l'apprenant d'effectuer des actions ou des modifications au système et d'observer les conséquences de ses actions.

- **Hypermedia** : consiste en un réseau de noeuds et de liens ou les noeuds sont des contenant d'information et les liens sont des relations entre les noeuds ou les portions de noeuds qui permet la manipulation d'un ensembles d'information selon un processus associative et fortement interactive.

1.2.2.2 Module expert / Modèle du domaine

Ce module représente les connaissances de la matière ou l'expertise du domaine. Les connaissances sont représentées par des modèles pouvant permettre le raisonnement sur ces dernières. Quelques exemples de modèle de représentation des connaissances sont les suivante :

- Règles de production (If - Then)
- Règle de production avec mesure de l'incertitude
- Réseaux sémantiques
- Représentations basées sur les objets structurés (frames)
- Les ontologies

Il exist 3 possibilités de modélisation :

- **Boîte noire** : appliquer une quelconque méthode de raisonnement sur le domaine, non explicitée, sans aucune transparence à l'utilisateur ;
- **Système expert** : peut expliquer ses raisonnements
- **Modèle cognitif** : peut simuler la façon dont l'humain utilise les connaissance.

1.2.2.3 Module de l'apprenant

Il représente l'état courant de l'apprenant. L'évaluation de cet état s'apparente à un processus de diagnostic médical.

Ce module est consulté périodiquement par le tuteur et l'expert pour déterminer le focus de la formation. Ce module est constitué de plusieurs sous modules à savoir :

- **Modèle cognitif** : qui est la plus part du temps de type superposition (overlay) c'est-à-dire que les connaissance de l'apprenant sont considérées comme formant un sous ensemble de celles de l'expert. Il existe 3 types de superpositions qui sont : *simple*, *perturbation*, *différentielle*. les approches pour implémenter les superpositions sont :
 - * Réseau sémantique
 - * Banque de connaissances de l'expert

- * Ensemble de compétences acquises par l'étudiant
- * Réseau bayésien
- **Modèle affectif** : permet le stockage des préférences de l'étudiant, de son profil psychologique, de son état émotionnel ou motivationnel. Ces informations peuvent varier selon le contexte et peuvent aussi évoluer dynamiquement. Le tuteur utilise ces informations pour adapter l'interaction avec l'étudiant
- **Modèle inférentiel** : qui permet de propager une évidence dans le modèle tout entier en s'assurant que le modèle reste cohérent après ces inférences.

1.2.2.4 Module du tuteur

Le tuteur contient les stratégies pédagogiques. Il doit choisir et planifier les activités à présenter à l'étudiant, lui fournir des explications adaptées et déterminer quand et comment intervenir.

Il existe plusieurs stratégies possibles qui définissent la situation dans laquelle aura lieu l'apprentissage comme par exemple :

- Entraînement(Coaching) : offrir à l'apprenant des conseils et le guider lorsqu'il s'éloigne de la solution ;
- Enseignement ou apprentissage socratique : articulation sur les éléments de connaissances du domaine pour amener à identifier les règles de niveau supérieur et les concepts ;
- Apprentissage exploratoire ;
- Apprentissage par perturbation ;
- etc...

Le tuteur s'appuie sur des approches éducatives appropriées pour prendre ses décisions.

1.3 Outils de modélisation des STI

1.3.1 Les ontologies

1.3.1.1 Définition

Une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations. Elle peut être vue comme un modèle de données représentatif d'un ensemble de concepts propres à un domaine, ainsi que des relations entre ces concepts. Selon

GRUBER (1993), “ l’ontologie est la spécification formelle et explicite d’une conceptualisation partagée d’un domaine”.

1.3.1.2 Représentation des ontologies

Pour la représentation des ontologies, on utilise principalement des langages formels dédiés. Ces langages doivent donc offrir des structures de données adaptées à la représentation des concepts. On distingue donc :

- **Les langages d’échange d’ontologies sur le web** : qui sont basés sur le langage XML ;
- **Les langages opérationnels** : qui sont utilisés lorsqu’on souhaite créer une ontologie constituée d’un composant d’un système d’information. Ce type de langage permet d’implémenter des ontologies à des fins d’inférence.

Pour formaliser les ontologies, on se sert de 4 types de composants à savoir :

- Concepts ;
- Relations : permettent de percevoir la structuration des concepts ;
- Axiomes : constituent des assertions, acceptées comme vraies ;
- Instances : qui véhiculent les connaissances à propos du domaine du problème

1.3.1.3 Types d’ontologies

- **Les ontologies génériques** : ce sont les ontologies qui définissent les concepts de façon générale et abstraite tels que l’espace, le temps et la matière. Les concepts définis à ce niveau peuvent être utilisés dans plusieurs domaines ;
- **Les ontologies du domaine et de tâches** : ce sont les ontologies qui décrivent les concepts spécifiques à une tâche ou à un domaine précis. Elle en donne une représentation formelle ainsi que les relations qui les lient ;
- **Les ontologies applicatives** : ce sont celles qui donnent une description des concepts dépendant à la fois d’un domaine spécifique et d’une tâche particulière. Il s’agit principalement de la spécialisation d’ontologies relatives ;
- **Les ontologies de représentation** : ce sont les ontologies qui donnent une conceptualisation des primitives des langages de représentation des connaissances. C’est le cas par exemple d’une ontologie sur la modélisation des connaissances chirurgicales ou une ontologie sur le formalisme des graphes.

1.3.1.4 Méthodologie de construction des ontologies

Les ontologies se construisent suivant les phases ci-après :

- **La phase de spécification** : dans cette phase, il s'agit de mieux appréhender l'étendu et le but de l'ontologie qu'on veut construire et donc de mieux délimiter son domaine d'application ;
- **L'acquisition de connaissances** : ici, on s'agit d'identifier les concepts pertinents , d'éliminer et/ou réduire les redondances, se rassurer de la fiabilité des documents ;
- **La conceptualisation** : faire la conceptualisation revient à définir le glossaire des termes de l'ontologie qui seront utilisés ;
- **La diffusion** : cette phase concerne la mise en place et le déploiement de l'ontologie construite ;
- **L'utilisation** : cette phase concerne toutes les activités reposant sur la disponibilité de l'ontologie, c'est-à-dire l'annotation de ressources , la déduction de connaissances et l'aide à la décision ;
- **L'évaluation** : ici, il est question d'évaluer l'ontologie construite est relever ses performances.

1.3.1.5 Les ontologies dans le domaine médical

La médecine est un domaine où les connaissances sont descriptives, la plus part du temps en langage naturel et non formalisées. Il existe plusieurs projets qui ont pour but de formaliser les connaissances médicales, comme par exemple :

- **SNOMED (Systematized Nomenclature of Medicine)**[7] : est une classification multiaxiale (topographie, morphologie, fonction, organismes vivants, médicaments/produits chimiques, agents physiques, etc.) et multi-domaine. A l'intérieur de chaque axe, les éléments sont organisés suivant une structure hiérarchique. La classification d'un terme repose sur une décomposition de celui-ci en combinaison de termes appartenant à différents axes.
- **UMLS (Unified Medical Language System)**[3] : n'est pas à proprement parler un projet de constitution d'ontologie. Il fait plutôt partie de la famille des terminologies. Il s'agit de l'union raisonnée de plus de quarante terminologies biomédicales dont le thésaurus MeSH(Medical Subject heading), la classification internationale des maladies et la nomenclature SNOMED III. UMLS a été mis en place dans le but d'améliorer l'accès à l'information médicale à partir de sources diverses : bases de

données bibliographiques, bases de données d'enregistrements cliniques et bases de connaissances médicales

1.3.2 Les réseaux bayésien

1.3.2.1 Définition

Un réseau bayésien est un système représentant la connaissance et permettant de calculer des probabilités conditionnelles apportant des solutions à différentes sortes de problématiques.

La structure de ce réseau est la suivante : Un **graphe** dans lequel les **noeuds** représentent des variables aléatoires et les **arc** reliant ces dernières sont rattachés à des probabilités conditionnelles.

L'intérêt particulier des réseaux bayésiens dans notre cas est de tenir compte simultanément de connaissances a priori d'experts (dans le graphe) et de l'expérience de l'apprenant (expérience dans les données).

1.3.2.2 Procédé de construction d'un graphe bayésien

Pour construire un réseau bayésien il faut :

- Définir le graphe du modèle
- Définir les tables de probabilités de chaque variable, conditionnellement à ses causes

1.3.2.3 Inférence bayésienne

Cette inférence se base sur des propriétés mathématiques plus précisément sur l'utilisation des énoncés probabilistes, qui dans le cas général sont trouvés par des experts étudiant un système qui leurs ait connu. Les règles de logique de probabilités utilisées sont les suivantes :

1. **La règle de l'addition** : $p(A \cup B/C) = p(A/C) + p(B/C) - p(A \cap B/C)$
2. **La règle de multiplication** : $p(A \cap B) = p(A/B)p(B) = p(B/A)p(A)$
3. **Le théorème de bayes** : $p(A/B) = \frac{p(B/A)p(A)}{p(B)}$

1.3.2.4 Applications

Les réseaux de bayes peuvent servir à faire :

1. Apprentissage Automatique Plusieurs méthodes existent selon que :

- La structure du réseau est connue ou non (apprentissage de paramètres et apprentissage de structure)
- Les valeurs des attributs sont disponibles ou pas
- Il y a un réseau initial proposé par les experts ou non
- On respecte les principes de l'apprentissage bayésien ou non

2. Inférences

- Diagnostic : des effets vers la cause : chaînage arrière
- Causale : des causes vers les effets : chaînage avant
- Inter-causale : entre les causes et un effet connu
- Mixte : combine un ou plusieurs des types précédent

1.3.3 Natural Language Processing(NLP)

1.3.3.1 Description

Le Natural Language Processing(NLP) ou Traitement Numérique du Langage est une discipline qui porte essentiellement sur la compréhension, la manipulation et la génération du langage naturel par les machines. Ainsi, le NLP est réellement à l'interface entre la science informatique et la linguistique.

Ce NLP va permettre de transformer la question de l'apprenant en question d'interrogation d'une ontologie. Pour ce faire, nous avons besoin d'extraire le nécessaire afin d'interroger l'ontologie. Cette situation nous mène plus vers l'extraction des mots clés.

En outre, nous avons besoin de générer une réponse pour l'apprenant à partir du résultat de l'ontologie. Ainsi, nous allons utiliser le NLP pour 2 cas de figures principaux à savoir :

- Traitement du texte de l'apprenant lors de sa discussion avec le patient virtuel : ou nous distinguons le NLU et OntoNLQA ;
- Génération de texte pour la réponse du patient virtuel à l'apprenant (NLG)

1.3.3.2 Natural Language Understanding(NLU)

Permet l'extraction des mots clés : Pour la réalisation de notre tâche qui consiste en l'extraction des mots clés, dans la littérature, il existe des modèles d'apprentissage automatique qui permettent de faire l'extraction des mots clés tel que les "Word Embedding", BERT.

1.3.3.3 OntoNLQA

C'est un framework basé sur le traitement automatique du langage qui permet d'interroger une ontologie à partir du langage naturel. Ce framework interroge une ontologie en utilisant SPARQL.

Il fonctionne comme suite

- La question(texte) passe au module est prétraité en appliquant la tokenisation, l'identification des mot racines, suppression des ponctuations à l'aide de la librairie Stanford CoreNLP.
- La prochaine étape consiste à faire la reconnaissance des entités en utilisant les "BertEmbedding" à l'aide de la librairie sparknlp. Les entités utilisées sont les classes et propriétés des ontologies. Identification des relations notamment à travers les sujets, verbes et complètement identifiés à l'étape précédente.
- Enfin, ces classes sont utilisées pour interroger l'ontologie à l'aide du langage SPARQL.

1.3.3.4 Natural Language Generation(NLG)

Ce module va nous permettre de générer la réponse à afficher à l'apprenant a partir du résultat de l'ontologie. Pour la génération des ces réponses, nous avons des modèles tels que GPT-3. Le modèle prend en entrée le résultat de l'ontologie et les entités identifiées précédemment dans le module OntoNLQA comme contexte et génère la réponse à afficher a l'apprenant.

Pour la réalisation de notre projet, nous allons utiliser OntoNLQA vu sa précision et sa simplicité par rapport aux méthodes d'extraction de mot clés qui vont permettre d'interroger l'ontologie.

1.3.4 Généralité sur la 3D

1.3.4.1 Définition

La synthèse d'images tridimensionnelles souvent abrégée 3D (3D pour trois dimensions : x, y, z, les trois axes qui constituent le repère orthonormé de la géométrie dans l'espace) est un ensemble de techniques notamment issues de la CAO(Conception Assisté par Ordinateur) qui permet la représentation d'objets en perspective sur un moniteur d'ordinateur.

1.3.4.2 Généralités sur les logiciels de modélisation tridimensionnelle

Un logiciel de modélisation tridimensionnelle, ou modelleur 3D, est un logiciel qui sert à créer des scènes 3D, composées de formes complexes, ou objets, en trois dimensions à partir de primitives de bases ou de définition analytique. Les modelleurs 3D sont utilisés aussi bien dans l'industrie en conception assistée par ordinateur que par les infographistes qui réalisent des scènes dédiées aux jeux vidéo, à la réalisation d'animations pour le cinéma et à la création de présentations ou d'environnements de réalité virtuelle.

Les logiciels de modélisation 3D se basent essentiellement sur la manipulation de formes de base. Ces formes de base utilisées peuvent être des cubes, des sphères ou des cônes, mais aussi des courbes de Bézier ou des NURBS, on parle alors de modelleur surfacique, à l'inverse des modelleurs polygonaux. L'utilisateur peut en ajouter ou en enlever à volonté.

Le logiciel propose généralement un ensemble d'outils qui permettent de modeler les formes de base afin d'obtenir des formes plus complexes, comme une voiture ou un personnage. Ces outils de modélisation peuvent être de simples transformations géométriques, et de la géométrie de construction de solides ou peuvent réaliser des transformations plus complexes, permettant de modifier des morceaux de la forme, de les découper ou de les tordre dans tous les sens. Les logiciels de modélisation 3D peuvent intervenir sur d'autres attributs comme la texture de l'objet, sa couleur, la manière dont il interagit avec la lumière, etc.

1.3.4.3 Quelques logiciels de modélisation tridimensionnelle

Il existe de nombreux logiciel de modélisation 3D parmi lesquels on peut citer :

- AutoCAD
- Blender
- SolidWorks
- SketchUP
- Etc...

1.4 Quelques SADM

Le diagnostic médical semble s'adapter difficilement aux contraintes de l'informatique, si l'on en juge par le petit nombre de programmes existant actuellement. Parmi les logiciels spécialisés dans l'assistance à la pose du diagnostic médical, nous présentons quelques uns.

1.4.1 ADM (Aide au Diagnostic Médical)

1.4.1.1 Contexte de réalisation

Le LIM (Laboratoire d'Informatique Médical) de la Faculté de Médecine de l'Université de Rennes a pour thème principal de recherche la modélisation des connaissances et la mise en œuvre des serveurs d'informations et de connaissances. Les projets et les applications développés concernent : L'aide à la décision diagnostique et thérapeutique, la modélisation numérique et conceptuelle des objets médicaux, le traitement du langage médical et des nomenclatures, l'enseignement en ligne, le campus numérique et l'université virtuelle. C'est dans ce contexte de recherche que le LIM a développé ADM (Aide au Diagnostic Médical) dont les objectifs consistent à aider les médecins à poser les diagnostics et à leur offrir un accès rapide à l'information médicale.

1.4.1.2 Les points de vue adoptés

Les concepteurs de ADM ont choisi de gérer les connaissances par un SGBD relationnel. Ce SGBD comporte des tables relatives aux descriptions de maladies, effets indésirables de médicaments, intoxications, des tables relatives aux entités descriptives, et aux mots. Il est régulièrement mis à jour et testé par les médecins qui alimentent la base. Le système inclut plusieurs fonctionnalités dont : l'évocation diagnostique, la stratégie de demande d'exams complémentaires, la prise en compte de circonstances particulières comme la grossesse ou l'allaitement, le support documentaire.

1.4.1.3 Les limites de ADM

Le logiciel ADM présente les limites suivantes :

- La gestion de la base de connaissance par un SGBD relationnel qui ne prend pas en compte les données complexes à cause de la première forme normale ;
- Les probabilités affectées aux diagnostics ne sont pas optimales et restent à améliorer ;
- Le système ne prend pas en compte les spécificités des signes dans les maladies ;
- Il n'y a pas un environnement virtuel qui permettrait à un apprenant de se sentir un peu dans les conditions réelles d'exercices de ses fonctions.

1.4.2 VIPS (Virtual Internet Patient Simulation)

1.4.2.1 Contexte de réalisation

Tout comme des simulateurs de vols, des simulateurs médicaux existent également, mais ceux-ci concernent principalement des aspects techniques bien définis (par exemple les simulateurs chirurgicaux). Le raisonnement clinique (et par conséquent la gestion des erreurs potentielles) n'est exploré que par quelques simulateurs, dont VIPS. VIPS est un simulateur de consultation médicale qui permet de tester et d'entraîner les capacités opérationnelles des soignants en utilisant une approche semblable à celle des pilotes de lignes.

1.4.2.2 Les objectifs de VIPS

L'objectif de VIPS est de parfaire la capacité d'un praticien à poser les questions pertinentes au malade et à interpréter les réponses pour prendre une décision. VIPS fournit des informations à l'apprenant uniquement en réponse à ses demandes et nécessite l'emploi du langage naturel pour poser des questions. VIPS évalue la performance de l'apprenant de façon automatique et permet de garder et d'analyser toutes les démarches effectuées lors de la consultation virtuelle. Lors de l'ouverture d'une session dans VIPS l'utilisateur sélectionne un cas qui est en fait la plainte du patient. Chaque consultation consiste en une série de démarches de l'utilisateur (interrogatoire, examens, décisions) qui peuvent être individuellement associées à une ou plusieurs hypothèses diagnostiques. L'utilisateur formule une question ou une hypothèse en langage naturel. Le logiciel interprète la question par recherche directe dans la base de données. L'utilisateur valide une interprétation du logiciel ou recommence l'opération.

1.4.2.3 Les limites de VIPS

- Le malade virtuel n'est pas réaliste ; il ne présente pas d'émotions qui peuvent caractériser ses souffrances ;
- L'interaction avec l'utilisateur est quasi inexistante ;
- L'interprétation du langage naturel peut fausser les résultats ;
- Pas d'environnement 3d, mais 2d, avec du simple texte.

1.4.3 SYDIME

SYDIME est un système développé au LIMSS, dans le cadre de ses travaux sur l'intelligence artificielle, la formation à distance et les programmes éducatifs. Il a pour objectif de définir une ontologie, afin de réaliser la base des connaissances qui sera exploitée par la

suite pour décrire les cas de maladies. Ne disposant d'une illustration du patient virtuel ce dernier a été amélioré en définissant un malade virtuel en 3D et animé. Ce dernier a été renommé en **MV-SYDIME** après l'ajout du malade virtuel.

1.4.3.1 Avantages

- L'utilisation des ontologies, comme dit plus haut les ontologies est l'outil le mieux adapté pour la représentation des connaissances dans le domaine médical, de plus l'inférence y est plus performante.
- La modélisation d'un patient virtuel permettant de représenter le malade à l'apprenant

1.4.3.2 Limites

- L'IHM du patient n'est pas très attractive
- L'apprenant ne peut pas consulter physiquement le patient c'est-à-dire que l'apprenant ne peut pas toucher une partie du corps du patient et il réagit
- Ne dispose que de l'évaluation sommative de l'apprenant.

MÉTHODOLOGIE

Dans cette partie nous allons présenter notre contribution dans la résolution du problème. Nous présentons une analyse, une conception, une modélisation, une architecture du système, une implémentation et enfin un déploiement de la solution.

Sommaire

2.1	Analyse	18
2.1.1	Définition de quelques concepts	18
2.1.2	Exigences du système	18
2.1.3	Diagrammes d'analyse	19
2.2	Conception	24
2.2.1	Diagramme de séquence technique de l'évaluation d'un apprenant	25
2.3	Modélisation	26
2.3.1	Interface	26
2.3.2	Modèle de l'apprenant	28
2.3.3	Modèle du tuteur	33
2.3.4	Module expert et décision	37
2.4	Architecture du système	39
2.4.1	Architecture globale	39
2.4.2	Architecture détaillée	41
2.5	Implémentation	41
2.5.1	Choix des outils	41

2.6	Déploiement	46
-----	-----------------------	-----------

2.1 Analyse

2.1.1 Définition de quelques concepts

- **Patient virtuel interactif** : est un simulateur informatique interactif destiné à la formation ou l'évaluation des professionnels de santé. Les patients virtuels reproduisent des situations cliniques, souvent issues de situations réelles. L'apprenant (un personnel de santé) interagit avec le patient virtuel dans le but de déterminer la maladie dont souffre le patient : poser un diagnostic
- **Diagnostic** : démarche par laquelle le personnel de santé détermine la maladie dont souffre un patient. Le processus de diagnostic se déroule en plusieurs étapes : le motif de la consultation, l'histoire de la maladie, les antécédents, les examens physiques, les hypothèses diagnostiques et les examens complémentaires.
- **Cas (de diagnostic)** : représente un cas de maladie porté par un patient virtuel. Il est constitué des *données démographiques* (données sur l'identité du patient), les *données cliniques* (données obtenues sur l'observation directe du patient) et les *données complémentaires* (questions, réponses et médias liés à la consultation du patient)

2.1.2 Exigences du système

Le système devra être capable de :

- offrir deux modes d'utilisation aux apprenants : mode d'apprentissage et mode évaluation
- présenter le patient virtuel émotif sous forme d'image 3D
- déterminer le niveau de l'apprenant et adapter les cours en fonction
- corriger l'apprenant

L'apprenant à travers le système doit être capable de :

- consulter un patient virtuel (poser des questions, prescrire un examen, donner le diagnostic)
- consulter ses notes
- consulter son parcours
- dialoguer avec le système

L'enseignant est chargé de :

- proposer des questionnaires et des évaluations par rapport à un thème ou un module
- créer et gérer les modules (ensemble de cas et de questions liés à une notion particulière par exemple diagnostic des maladies infectieuses)
- Faire des commentaires sur un cas (commentaires qui permettront à l'apprenant de mieux comprendre la solution)
- consulter les notes de l'apprenant

L'expert est chargé de :

- introduire de nouveaux cas dans le système
- valider les cas

2.1.3 Diagrammes d'analyse

2.1.3.1 Diagramme de contexte

Le diagramme suivant présente les différents acteurs du système :

- **Apprenant** : (acteur primaire) c'est le personnel de santé qui utilise le système enfin de se former au diagnostic médical
- **Enseignant** : (acteur secondaire)
- **Expert** : (acteur secondaire)

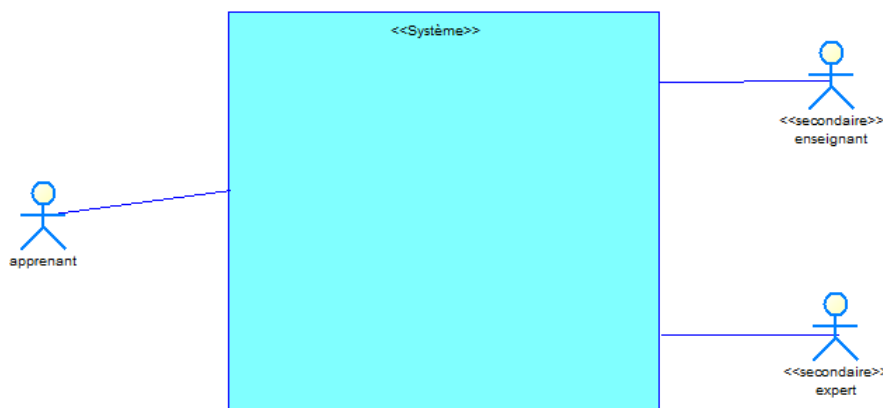


FIGURE 2.1. Diagramme de contexte

2.1.3.2 Diagramme de cas d'utilisation

Ce diagramme présente toutes les interactions entre les acteurs et le système.

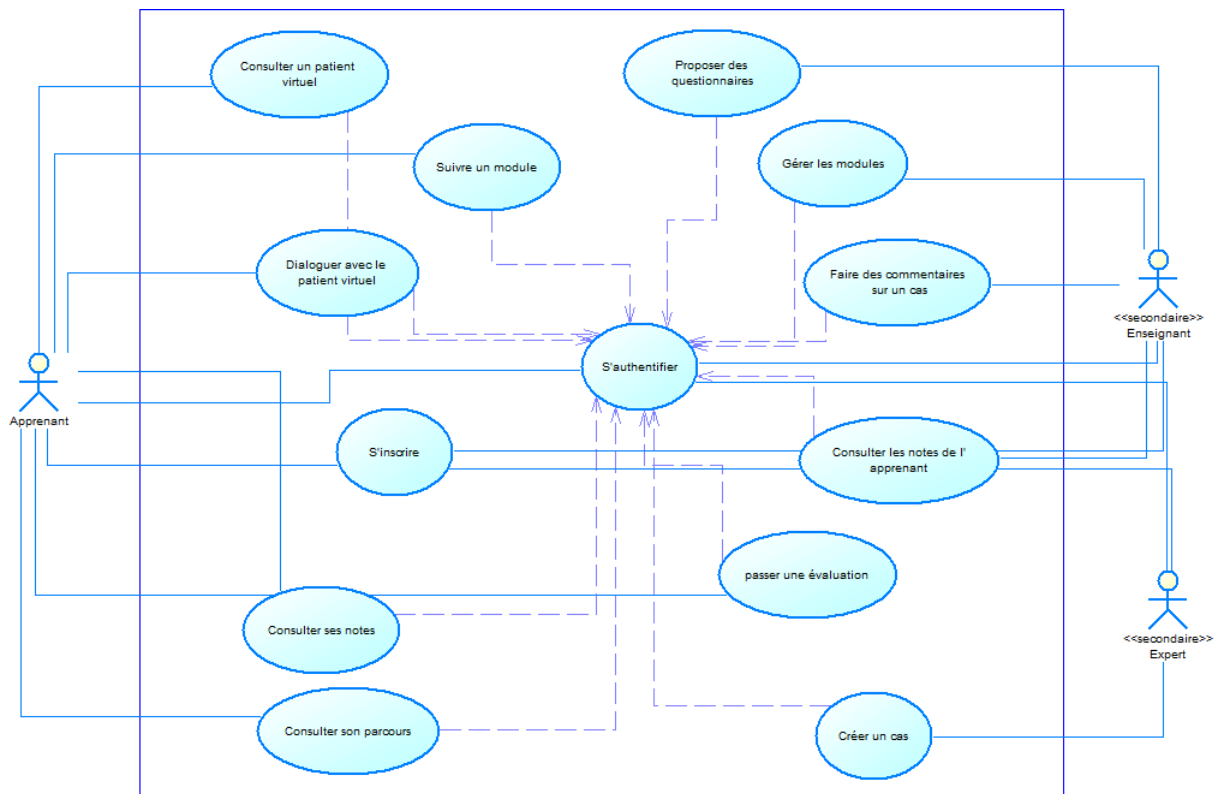


FIGURE 2.2. Diagramme de cas d'utilisation

Acteur	Type	Cas d'utilisation
Apprenant	Primaire	<ul style="list-style-type: none"> — S'inscrire — S'authentifier — Consulter un patient virtuel — Suivre un module — Dialoguer avec le patient virtuel — Consulter ses notes — Consulter ses parcours — Passer une évaluation
Enseignant	Secondaire	<ul style="list-style-type: none"> — S'inscrire — S'authentifier — Proposer des questionnaires — Gérer les modules — Faire des commentaires sur un cas — Consulter les notes de l'apprenant
Expert	Secondaire	<ul style="list-style-type: none"> — S'inscrire — S'authentifier — Créer un cas

TABLE 2.1 – Cas d'utilisation Global

2.1.3.3 Diagramme de classe métier

Ici nous allons vous présenter les différentes entités que nous avons pu recenser pour mettre en place notre système. Ce diagramme présente de manière très brève les différentes entités sans leurs comportements.

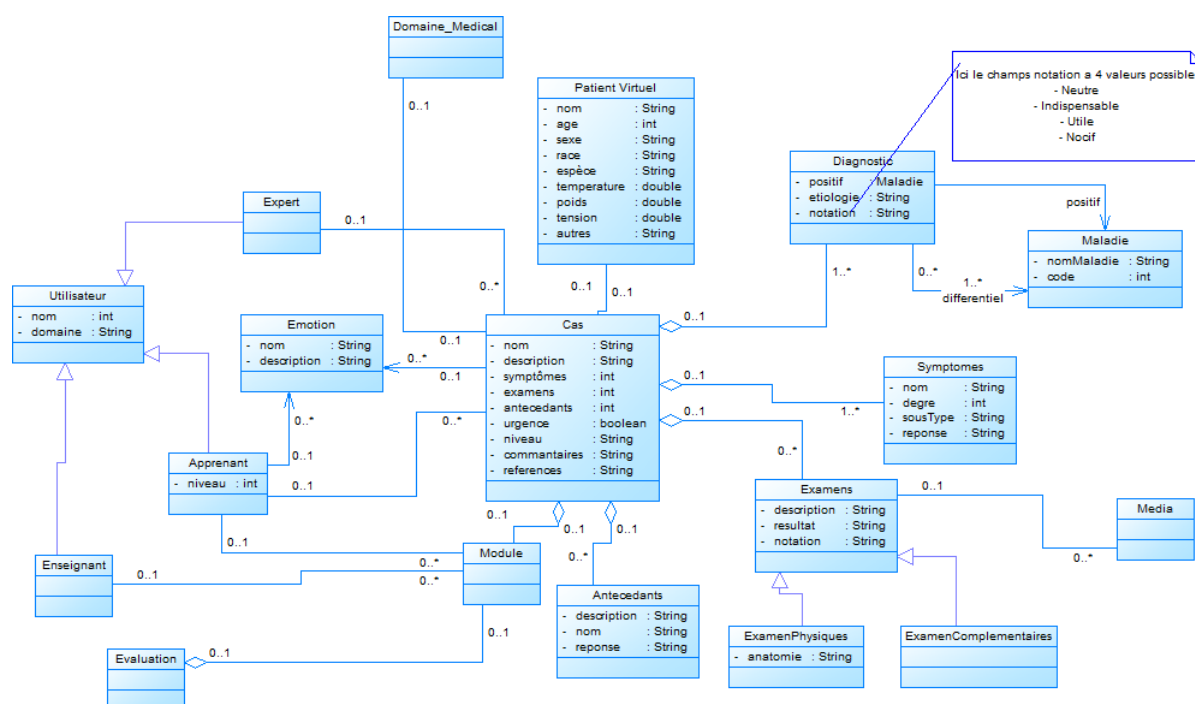


FIGURE 2.3. Diagramme de classe métier

2.1.3.4 Activité

Le diagramme d'activité quant à lui, représente les différentes tâches ou actions intervenant dans le processus de l'évaluation. Nous allons distinguer pour cela deux types d'évaluation à savoir :

- Evaluation Formative
- Evaluation Sommative

Evaluation formative

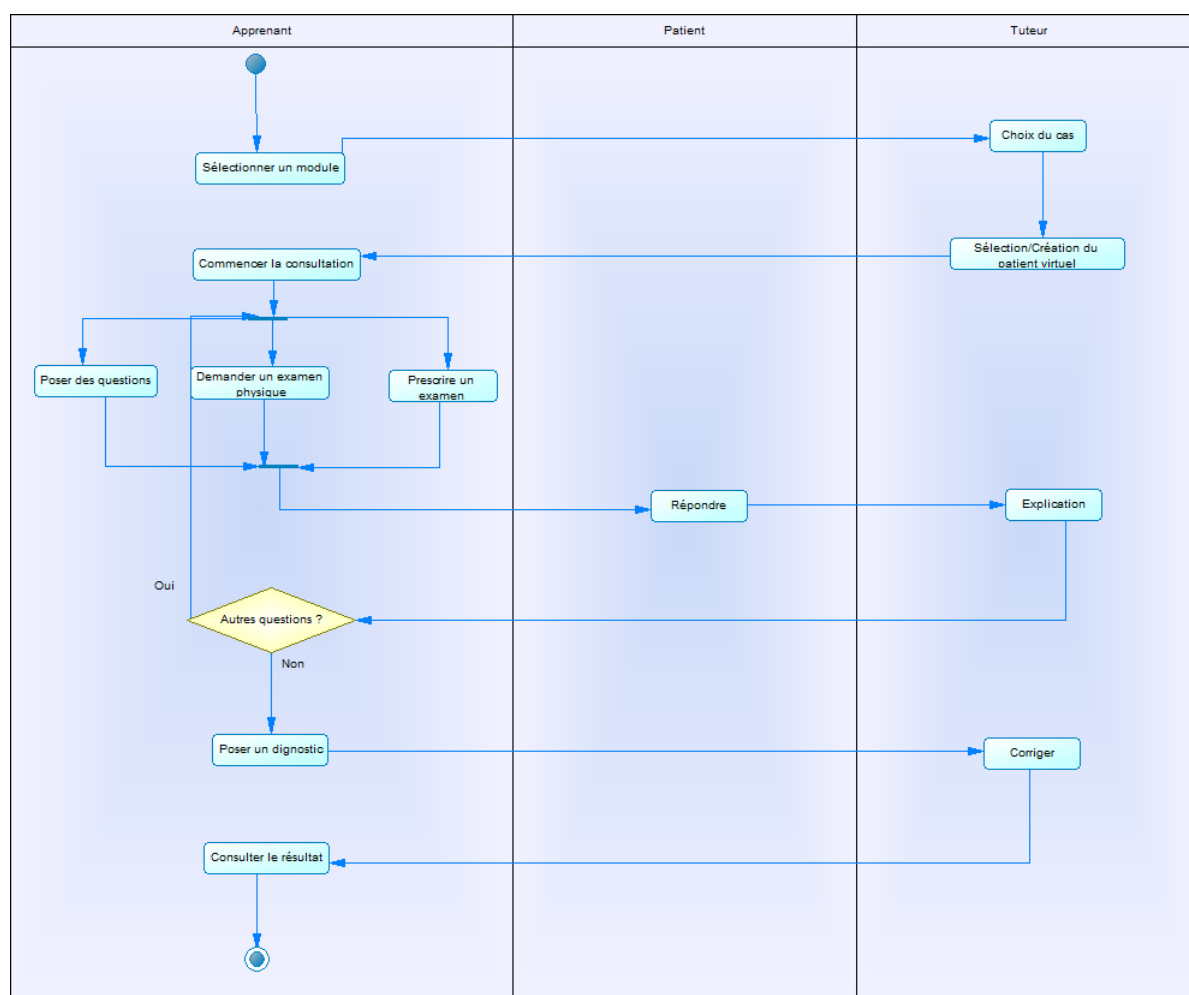


FIGURE 2.4. Diagramme d'activité du processus d'évaluation formatif

Evaluation sommative

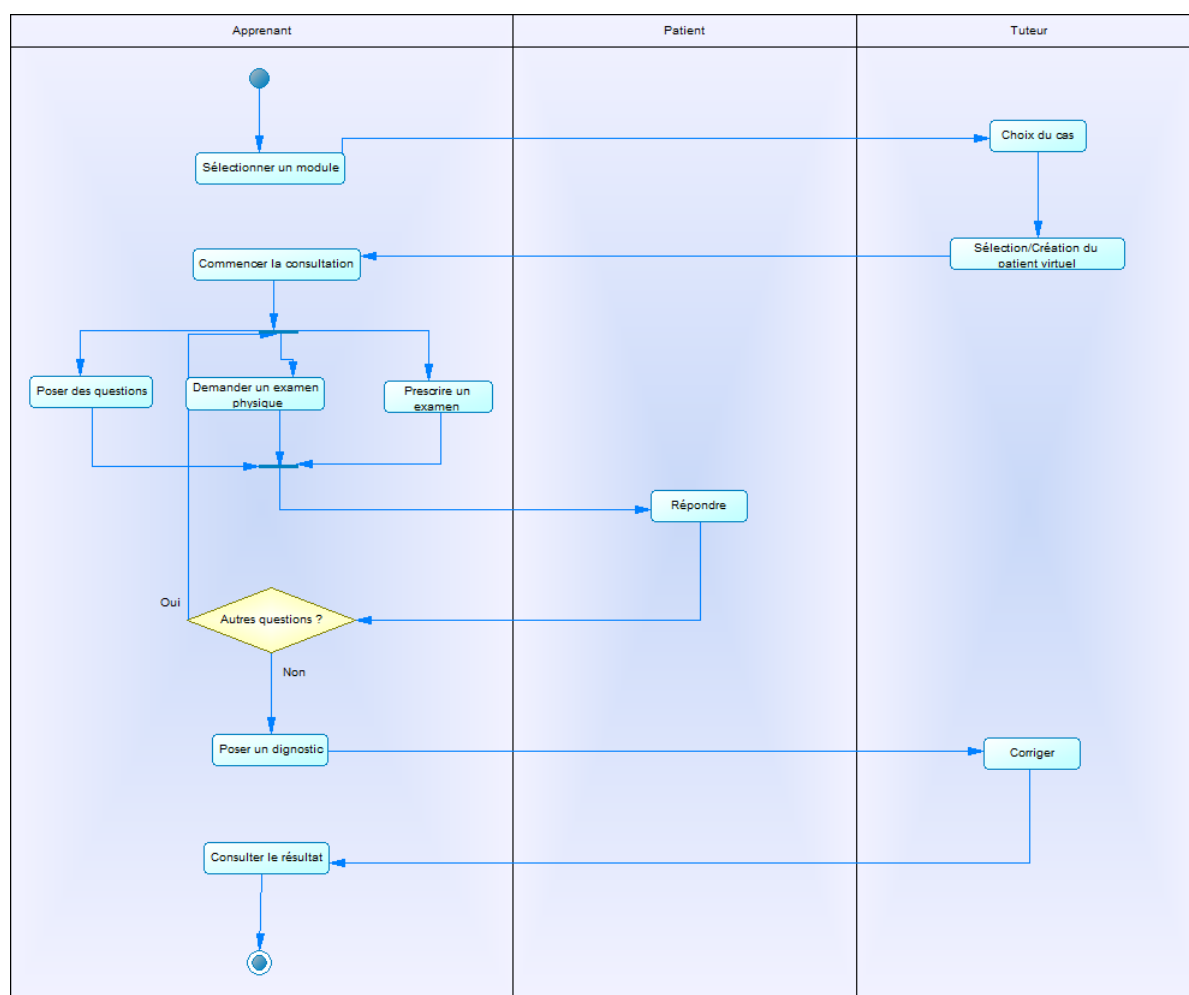


FIGURE 2.5. Diagramme d'activité du processus d'évaluation sommatif

2.2 Conception

2.2.1 Diagramme de séquence technique de l'évaluation d'un apprenant

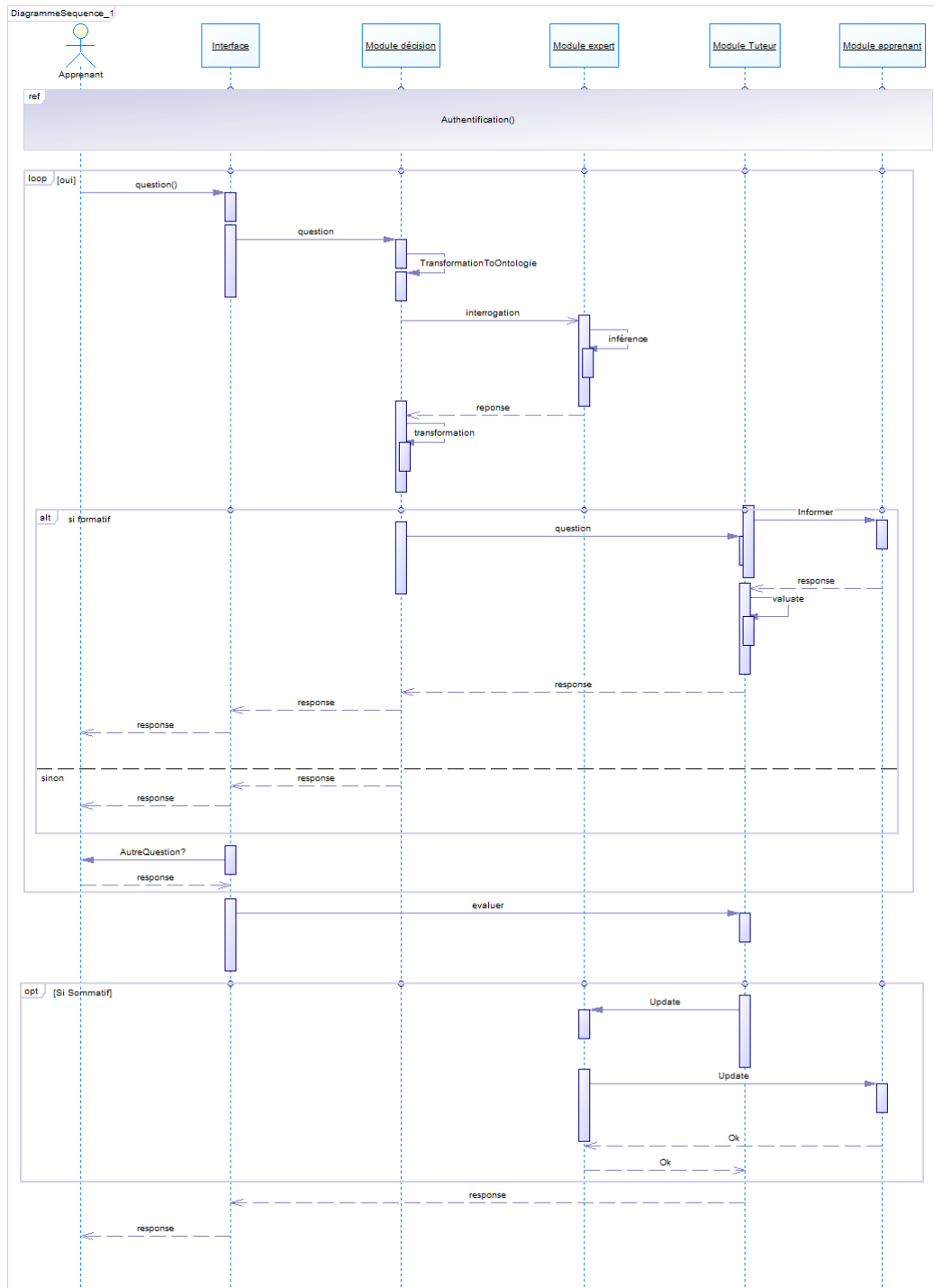


FIGURE 2.6. Diagramme de séquence technique de l'évaluation d'un apprenant

2.3 Modélisation

2.3.1 Interface

Pour la réalisation de ce module, nous allons utiliser une interface simulateur. La simulation va consister à faire une représentation virtuelle d'un patient qui permet de ressortir le comportement corporel d'un patient afin d'apporter plus d'informations à l'apprenant.

En outre, l'interface permettra à l'apprenant de poser des questions au patient et de poser des diagnostics qui seront entre sous forme de texte au système.

Quelques interfaces utilisateurs sont les suivantes :

- Simulation d'un patient virtuel avec les différentes options qui se présentent à l'apprenant.

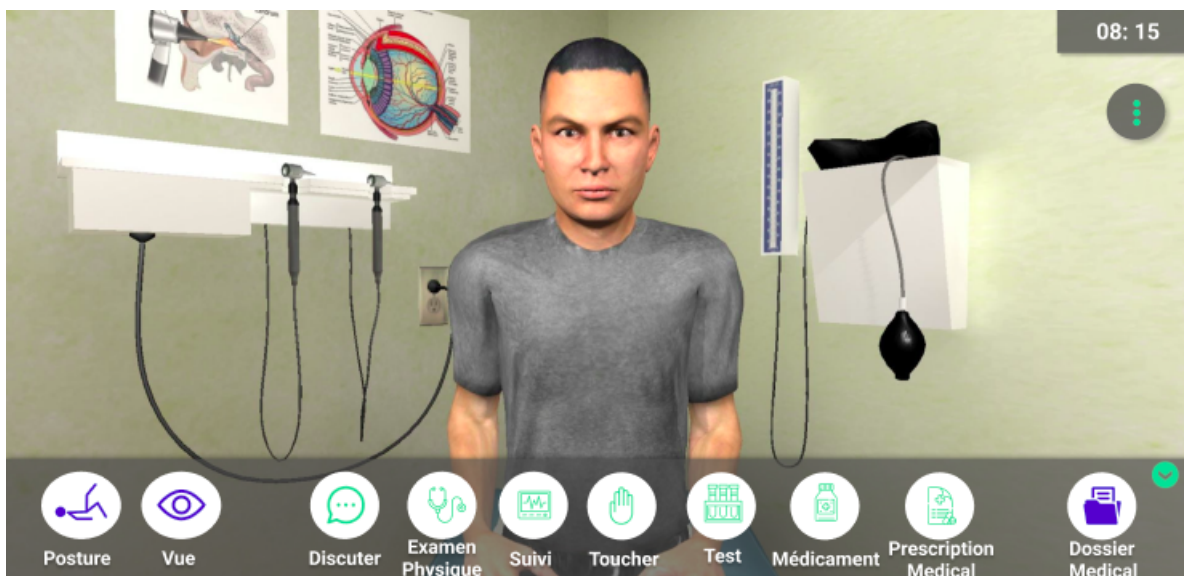


FIGURE 2.7. Interface avec différentes options

- Affichage du composant montrant les différents tests qu'on peut effectuer et un résultat de radiologie. Ces composant sont déplaçable

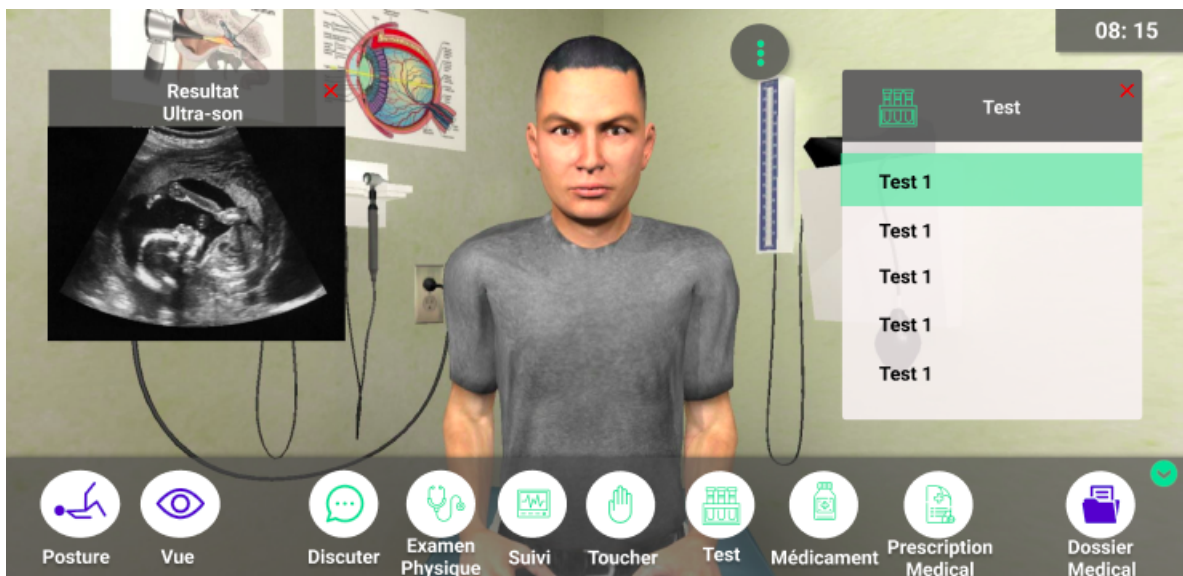


FIGURE 2.8. Affichage des composantes de test

- Main activer par l'apprenant et permet de simuler le toucher en faisant un click gauche de la souris, le patient virtuelle peut reagir en disant quelque chose qui est affiché comme notification en bas et a droite de l'ecran.

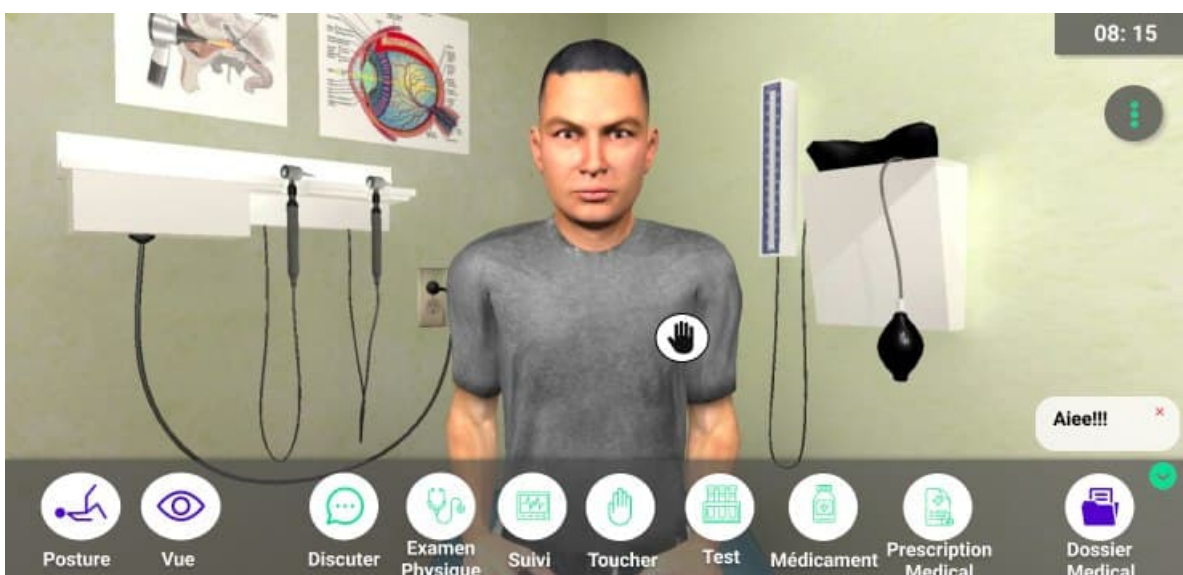


FIGURE 2.9. Simulation de la consultation physique

- montre une discussion avec entre l'apprenant le patient virtuelle.



FIGURE 2.10. Discussion entre l'apprenant et le patient virtuel

2.3.2 Modèle de l'apprenant

Il représente l'état courant des connaissances de notre l'apprenant et la recherche de cet état s'apparente à un processus de diagnostic médical. Ce module est consulté périodiquement par le tuteur et l'expert pour choisir de présenter les prochaines connaissances. il est constitué de trois sous modèle :

- Modèle cognitif
- Modèle affectif
- Modèle inférentiel

2.3.2.1 Modèle cognitif

Nous avons choisi le type **Superposition** ou **Overlay** dans lequel les connaissances des apprenants sont une sous partie de celles de l'expert.



FIGURE 2.11. Type Overlay

Nous caractérisons ce modèle par :

- **Le niveau de l'apprenant** : qui sera le moyen de connaître l'état des connaissances de l'apprenant dans l'apprentissage souhaité. On pourra le ressortir grâce à l'ensemble des connaissances de l'apprenant que nous allons matérialiser par un réseau bayésien.

Le développement d'un modèle de connaissance médical réaliste est généralement loin d'être facile, et l'utilisation de réseaux bayésiens ne fait pas exception à la règle. Comme c'est le cas pour d'autres formalismes de représentation, il existe des directives particulières qui facilitent le développement d'un réseau bayésien. Nous commencerons par résumer globalement la connaissance de notre apprenant.

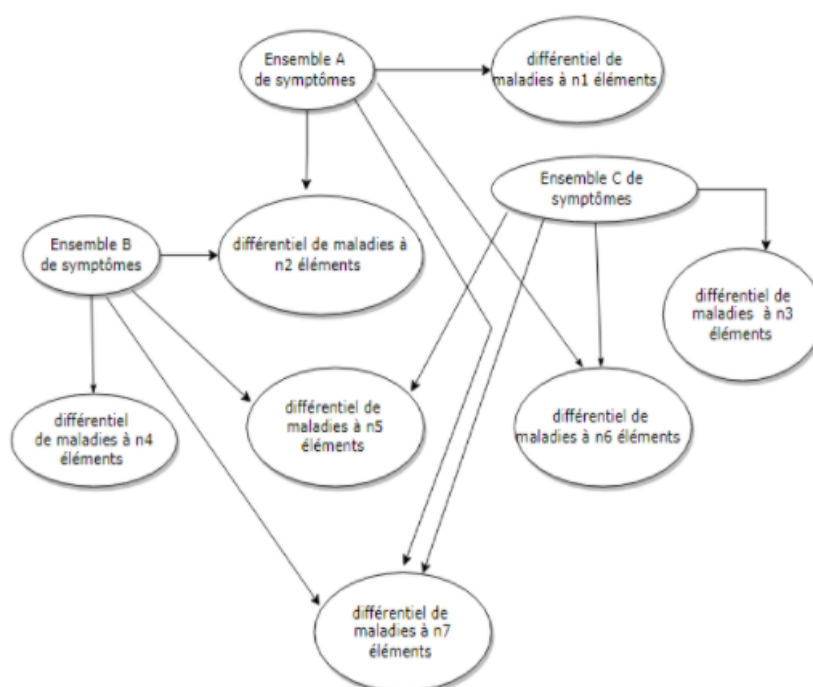


FIGURE 2.12. Structure globale de la connaissance de l'apprenant

Les arcs représentent les influencent stochastiques.

Modèle de baye

La figure ci-dessus donne une vue globale de l'ensemble des connaissances de l'apprenant. La structure d'un réseau de baye peut être conçue en utilisant la connaissance des dépendances causales, influences ou corrélations causales connues. Tout ou partie de ces connaissances étant dérivé des connaissances de l'expert, via le processus d'apprentissage.

Formellement un réseau de baye $B=(G, Pr)$ est un graphe orienté acyclique $G=(V(G), A(G))$ avec un ensemble de sommets $V(G) = V_1, \dots, V_n$ représentant des variables stochastiques, et un ensemble d'arcs $A(G) \subseteq V(G) \times V(G)$, représentant les dépendances et indépendances stochastiques entre les variables. Sur l'ensemble des variables stochastiques une distribution de probabilité conjointe $Pr(V_1, \dots, V_n)$ est définie qui respecte les dépendances et indépendances représentées dans le graphe $Pr(V_1, \dots, V_n) = \prod_{k=1}^n Pr(V_k | \prod(V_i))$, où $\prod(V_i)$ représente les variables correspondant aux parents du sommet V_i .

Réseau de baye des connaissances de l'apprenant

- Graphe acyclique : les nœuds aléatoires sont les maladies , et les différentes dépendances entre nœuds sont les différents symptômes commun à chaque maladies.
- Distribution de probabilité : chaque nœud ou maladie sera caractérisé par :
 - * La probabilité que cette maladie soit connue de l'apprenant et qui représentera le niveau de l'apprenant pour cette maladie, cette probabilité sera mise à jour par le tuteur.
 - * Le niveau d'ancienneté qui sera le rang pendant lequel la maladie aurait été apprise pour la première fois, il y a une période T qui définit le délai pendant lequel les maladies apprises sont de même rang d'ancienneté.
 - * Toute probabilité non nulle qu'une autre maladie apprise directement plus tôt que(c'est à dire de niveau d'ancienneté actuel -1) elle soit connue.

Ainsi, chaque nœud aura une table de 2 colonnes et (n+2) ligne pour conserver ses valeurs. n est le nombre de maladies directement connues(c'est-à-dire de niveau d'ancienneté actuel -1) avant la maladie.

- Organigramme de création du réseau

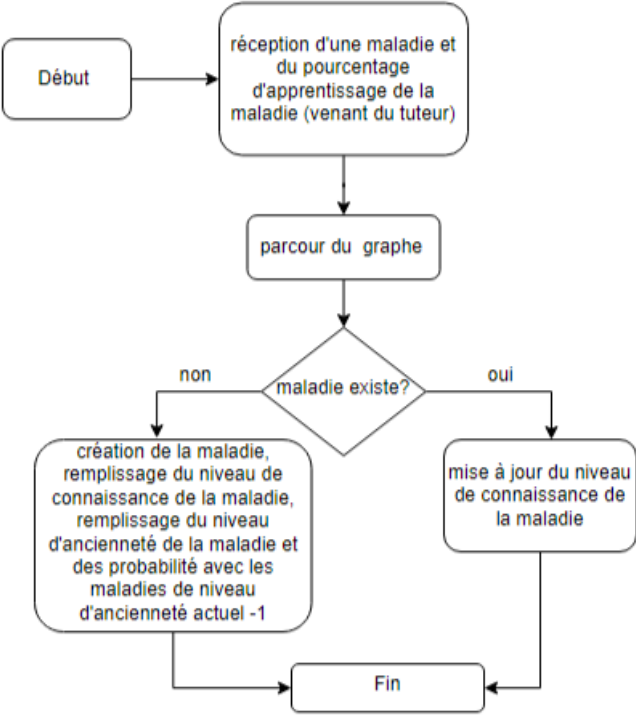


FIGURE 2.13. Organigramme de création du réseau Bayésien

— Exemple d'état à un instant donné

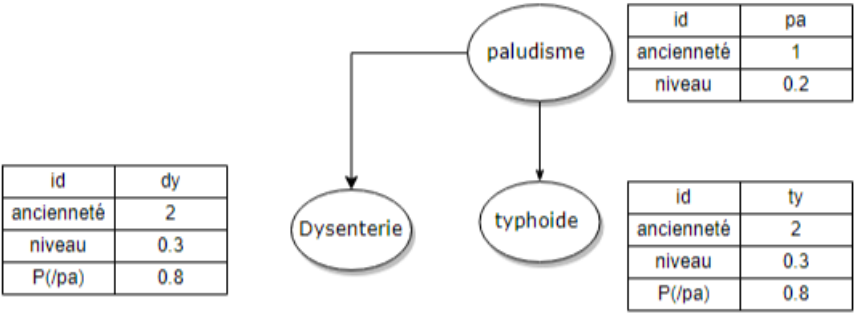


FIGURE 2.14. Exemple de connaissances

2.3.2.2 Modèle affectif

Nous caractérisons l'impact des émotions sur sa faculté cognitive.

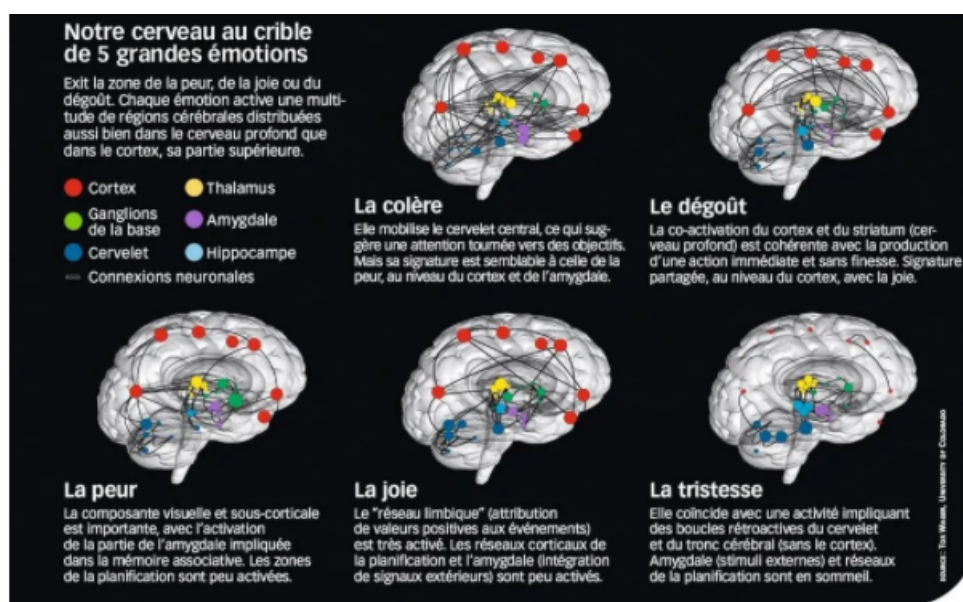


FIGURE 2.15. Recette émotionnelle

Nous attribuons donc à chacune de ces principales émotions un pourcentage d'impact positif ou négatif sur la cognition, cette fonction est implémentée au niveau du tuteur. La détection de l'émotion actuelle de l'apprenant sera faite au niveau de l'interface dans ce sens où elle permet de reconnaître les expressions faciales des apprenants afin de détecter leurs émotions. Cela permet de détecter l'état de l'apprenant avant le début d'un cours ou d'une évaluation. Est-il en colère, triste, heureux, nerveux, anxieux...

Ici nous utiliserons the haar like cascade algorithm, c'est un algorithme de deep learning utilisant les réseaux de neurones convolutifs (CNN) et la librairie célèbre de traitement d'image **OpenCV**.

2.3.2.3 Modèle inférentiel

Nous allons utiliser le **knowledge tracing** plus de l'**inférence** pour mettre à jour l'état des connaissances de l'apprenant. Le **knowledge tracing** permet d'analyser une épisode d'apprentissage afin d'identifier les connaissances qui ont été utilisées et un avantage en plus est qu'il ne nécessite pas une modélisation sophistiquée du processus de résolution de problème.

De plus, les réseaux bayésiens nous donnent la possibilité de mettre à jour automatiquement les informations au niveau des nœuds c'est-à-dire mettre à jour les probabilités.

2.3.3 Modèle du tuteur

Il est représenté par des modèles pouvant permettre de raisonner sur ces connaissances. Dans cette partie, nous allons définir la communication entre le patient et l'apprenant dans un premier temps, puis, nous allons présenter les stratégies pédagogiques utilisées, ensuite nous allons présenter les critères d'évaluation utilisés par le tuteur et enfin présenter le déroulement de l'enseignement.

2.3.3.1 Représentation des connaissances

Dans cette partie, nous allons montrer comment se représentent les connaissances de l'expert chez le tuteur, afin que ces dernières puissent aisément être manipulées et fournir un apprentissage de qualité à l'apprenant.

Nous allons opter ici pour une représentation des connaissances grâce aux réseaux bayésiens. Nous construisons le réseau en nous basant sur une hiérarchie des connaissances ; il contient toutes les connaissances liées au cas (celles permettant de manipuler les hypothèses et les évidences qui peuvent être faites lors du diagnostic d'une maladie).

Dans un premier temps, dans le réseau, des nœuds qui modélisent des hypothèses ; Chaque nœud possède une ou plusieurs hypothèses filles, et chaque fille ayant une mère dans le réseau. La structure hiérarchique dans le réseau bayésien est conservée, telle que se soit comme si c'est un expert qui a formulé ces hypothèses. Par exemple : On peut avoir gynécologie, hypothèse mère de infection gynécologique, MST, kyste ovarien, et kyste ovarien est mère de torsion de l'ovaire.

Les évidences sont représentées par des nœuds ; Une évidence peut influencer une ou plusieurs hypothèses ; et elles peuvent être obtenues par plusieurs moyens. Par exemple, si l'apprenant pose la question au patient : "Avez vous déjà subis des tests de dépistage pour les MST?" . cela pourrait apporter les évidences dépistages MST, méthodes barrières. La température du patient (qui est une évidence) peut être obtenue soit en demandant au patient directement, soit obtenue après avoir passé un examen physique. Ainsi, on remarque que tous les liens ne sont pas des relations de cause->effets, mais il y en a d' évidence -> hypothèse.

Afin de mieux expliquer le déroulement d'une évaluation formative, les éléments suivant sont à prendre en compte :

- **Vérification du choix des hypothèses** : pour savoir si l'étudiant est entrain de bien choisir ses hypothèses, le tuteur regarde l'hypothèse parmi les hypothèses celle qui sont assez probables ; de plus, il calcule l'influence des évidences sur l'hypothèse en

cours (en comparant les différentes probabilités) et il cherchera aussi les valeurs des noeuds parents influençant le plus l'hypothèse la plus probable pour pouvoir aboutir à un résultat tel que celui qu'un médecin l'aurait fait.

- **Validation du diagnostic** : pour obtenir un diagnostic similaire à celui de l'expert dans le module du tuteur, on regardera l'hypothèse la plus probable, car il s'agira sans doute de la plus précise. On notera cependant que si une hypothèse mère n'est pas valide, aucune hypothèse fille ne pourra l'être. On aura donc :

$$p(H_{fille}|!H_{mere}) = 0$$

Ainsi, la probabilité d'une hypothèse fille sera toujours plus petite que celle de son parent puisque :

$$p(H_{fille}) = p(H_{fille}|H_{mere}) * p(H_{mere}) + p(H_{fille}|!H_{mere}) * p(!H_{mere})$$

or $p(H_{fille}|!H_{mere})$ et $p(H_{mere})$ sont inférieurs à 1. On a finalement

$$p(H_{fille}) \leq p(H_{mere})$$

L'hypothèse à diagnostiquer est donc l'hypothèse la plus probable parmi les hypothèses terminales (les feuilles).

2.3.3.2 Gestion et calculs dans un graphe probabiliste avec la bibliothèque pyAgrum : Cas du graphe Bayésien

Dans cette partie, nous allons expliquer dans un premier temps ce que c'est que pyAgrum et ensuite comment l'utiliser pour faire des calculs d'inférence dans un graphe bayésien.

pyAgrum est une bibliothèque scientifique C++ et Python dédiée aux réseaux bayésiens (BN) et autres modèles graphiques probabilistes. Basée sur la bibliothèque C++ aGrUM, elle fournit une interface de haut niveau à la partie C++ d'aGrUM permettant de créer, gérer et effectuer des calculs efficaces avec les réseaux bayésiens et autres modèles graphiques probabilistes : réseaux de Markov (MN), diagrammes d'influence (ID) et LIMIDs, réseaux crédibles (CN), BN dynamiques (dBN), modèles relationnels probabilistes (PRM).

Plusieurs sujets ont été ajoutés à pyAgrum (en tant que modules purement python utilisant pyAgrum) :

- Classificateurs probabilistes conformes à Scikit-learn basés sur les réseaux bayésiens,
- Causalité probabiliste (réseaux causaux, do-calcul),

- Réseau bayésien dynamique,
- Outils pour l'explicabilité dans les réseaux bayésiens.

Pour pouvoir l'utiliser il suffit de l'installer.

2.3.3.3 Communication entre le patient et l'apprenant

Pendant la réalisation d'un diagnostic par un apprenant, il peut être amené à poser des questions au patient virtuelle, ceci est réalisé dans un processus décrit comme suit :

- L'apprenant pose une question au patient virtuelle en saisissant la question dans la zone de text de discussion avec le patient virtuelle.
- Le texte est envoyé de l'interface au serveur qui va traiter les documents comme suit :
Le texte est passé en entrée au module OntoNLQA qui permet d'interroger directement.

2.3.3.4 Stratégie pédagogique utilisée

Nous utilisons deux stratégies principales d'évaluation dans notre système : Une stratégie formative et Une stratégie sommative.

Stratégie formative

Dans ce type d'évaluation, le tuteur et l'apprenant seront appelés à échanger très fréquemment durant l'apprentissage. Une évaluation formative est similaire à un exercice d'entraînement; son but est de préparer l'apprenant à aborder les exercices sommatifs dont nous parlerons par la suite.

Lors d'un exercice formatif, l'apprenant émet ses hypothèses et continue de poser les questions aux patients en récoltant des évidences par les réponses de ce dernier. Quand il pose des questions pour changer d'hypothèses par exemple, le tuteur pourra intervenir pour demander à l'étudiant pourquoi il change? Sur quel critère s'est t-il basé pour changer ou alors pour émettre telle hypothèse. Au bout de cet exercice, aucun paramètre lié à l'apprenant n'est mis à jour. Un des paramètres de l'apprenant serait par exemple le niveau de l'apprenant. Son niveau ne va varier qu'au bout d'une évaluation sommative.

Stratégie sommative

L'évaluation sommative est un contrôle de connaissance que l'apprenant fait. Le tuteur intervient seulement à la fin de l'évaluation avec les réponses aux questions de l'évaluation.

Le niveau de l'apprenant à donner un diagnostic médical est influencé par la note obtenue à l'évaluation sommative.

2.3.3.5 Critères d'évaluation (dans un exercice sommatif)

L'apprenant n'obtient pas de note à la fin d'une évaluation formative ; puisque telle que définie, il s'agit d'exercices d'entraînement.

Une notation est également attachée à chaque élément (examen, diagnostic positif, diagnostic différentiel, diagnostic étiologique). Elle est neutre par défaut. Les diagnostics et les examens cliniques peuvent être indispensables, utiles, neutres ou nocifs. À l'issue de l'évaluation, le compte rendu personnalisé est généré automatiquement, comprenant les scores, les résultats à chaque phase (examen et les diagnostics que l'apprenant devait donner suivant leur notation selon ce que l'expert a renseigné), l'historique, un commentaire de l'auteur, des références bibliographiques. Trois autres scores en dehors de la méthode utilisée sont affichés : global, efficacité et efficience.

L'historique affiche toutes les actions de l'apprenant dans l'ordre dans lequel elles ont été effectuées, avec pour chaque action l'étape, le libellé, la réponse, le score de notation.

Le score global est calculé ainsi : demander un item utile ajoute des points (indispensable + 4 points ; utile + 2 points ; neutre + 0 points), demander un item néfaste retire des points (néfaste - 1 point ; excluant : zéro à l'étape de la consultation où l'item excluant a été donné).

$$Score_{global} = \frac{indisp.effectués \times 4 + utileseffectués \times 2 - néfasteseffectués \times 1}{totalutiles \times 2 + totalindispensable \times 4}$$

$$Efficience = \frac{indispensableseffectués + utileseffectués}{totaleffectués} \times 100$$

$$Efficacité = \frac{indispensableseffectués + utileseffectués}{totalutiles et indispensables} \times 100$$

2.3.3.6 Définition du niveau de l'étudiant

Le niveau de l'étudiant est entièrement déterminé par sa performance (efficacité) lors des évaluations sommatives.

Chaque cas à un niveau (facile, moyen ou difficile). Le système fixe pour chaque type de cas propre à une maladie, un nombre nf de cas facile, un nombre nm de cas moyen, et un nombre nd de cas difficile de telle sorte que :

$$\frac{nf}{100} + \frac{nm}{100} + \frac{nd}{100} = 1$$

Le système possède aussi pour chaque maladie les variables *sf*, *sm*, et *sd* qui sauvegardent les pourcentages déjà obtenus par l'apprenant respectivement pour les cas facile, moyen et difficile. Ainsi l'attribution du pourcentage d'évolution se fait par la fonction ***attribperc()*** qui suit :

Algorithm 1 *attribperc()*

```

if type(cas) == facile then
    note  $\leftarrow \frac{1}{100} \times \textit{efficacité}/x$ 
    if sf + note  $\leq \frac{nf}{100}$  then
        sf  $\leftarrow sf + note$ 
    return note
    end if
end if
if type(cas) == moyen then
    note  $\leftarrow \frac{1}{100} \times \textit{efficacité}/x$ 
    if sm + note  $\leq \frac{nm}{100}$  then
        sm  $\leftarrow sm + note$ 
    return note
    end if
end if
if type(cas) == difficile then
    note  $\leftarrow \frac{1}{100} \times \textit{efficacité}/x$ 
    if sd + note  $\leq \frac{nd}{100}$  then
        sd  $\leftarrow sd + note$ 
    return note
    end if
end if
return 0

```

2.3.4 Module expert et décision

Ce module représente les connaissances du domaine et moyen d'inferer sur celles ci. Tel que définit dans les diagrammes de séquence ci dessus, le module expert a pour objectif de :

- * **gérer les cas** : il s'agit de toutes les opérations de création, suppression, mise à jour et lecture d'un cas
- * **rechercher d'un cas** : consiste à rechercher un ou plusieurs cas en fonction d'un (ou de plusieurs) critère(s). Ceci permettra d'obtenir le cas adéquat à présenter à un apprenant en fonction de son niveau d'apprentissage et de sa spécialité.

- * **comprendre et répondre aux questions posées par l'apprenant** : afin de permettre un dialogue interactif entre le patient virtuel et l'apprenant

Par la suite nous présentons le modèle de représentation des connaissances et les algorithmes qui seront implémentés par le module expert pour répondre aux attentes du système.

2.3.4.1 Modèle de représentation des connaissances

Dans le cas du diagnostic médical, les connaissances que nous aurons à représenter sont les suivantes : cas, maladie, examen, symptômes, diagnostic, questions et les réponses associées, le patient, etc... tel que définit dans le diagramme de classe. Pour la modélisation de ces connaissances, nous utiliserons les **ontologies** qui sont recommandées pour la modélisation des connaissances médicales parce qu'elles fournissent un bon moyen de raisonnement sur celles ci.

- **De la construction de l'ontologie** : Etant donné qu'il existe déjà des ontologies adaptées au domaine médical, nous partirons sur la base des ontologies telles que *UMLS* et *SNOMED* mais restreint aux connaissances liés au diagnostic médical. A ces connaissances, nous ajouterons d'autres connaissances nécessaires au fonctionnement du système telles que : patient, apprenant.
- **De la création des instances**
- **De l'interrogation de l'ontologie** : elle permet l'inférence sur la base de connaissances. Pour ce faire nous utiliserons un langage de requête d'ontologie à l'instar de **SparQL**
- **De la persistance des instances** : une fois créées et manipulées, les instances devront être stockées enfin d'être réutilisées par la suite. Il est donc nécessaire de créer une base de données (relationnelles) qui permettent de stocker les instances créées et qui soit en adéquation avec l'ontologie créée.

Plusieurs recherches ont été faites à ce sujet (la persistance des instances d'une ontologie) enfin de savoir comment permettre le couplage entre les différents supports de stockage et l'interrogation de l'information. Parmi ces recherches, nous avons principalement :

- * **la conversion ontologie <-> base de données relationnelles** : elle permet de définir les règles pour quitter d'une ontologie vers une base de données relationnelles. IL existe aussi des outils pour le faire, tels que **CoDBOnto**

- * **l'utilisation de bases de données à base ontologique** : ce sont des bases qui permettent non seulement de gérer à la fois des ontologies et des données, mais aussi d'associer à chaque donnée le concept ontologique qui en définit le sens. Grâce à ces bases de données, il est possible de gérer une base de données comme si c'était une ontologie.

Au vu de tout ceci, nous avons choisi d'utiliser les **bases de données à base ontologique** car elles nous permettent à la fois de : construire l'ontologie, créer les instances, interroger et persister les instances.

2.3.4.2 Manipulation des connaissances

Pour la manipulation des connaissances, nous utiliserons :

- **Le langage de requête ontologique** : pour la gestion des cas. Il permet les CRUD (création, suppression, mise à jour, lecture) sur les données.
- **Le traitement du langage naturel (NLP)** : pour la compréhension et la réponse aux questions posées au patient virtuel par l'apprenant. Nous procéderons comme suit :
 - * extraire les mots clés de la question posée par l'apprenant
 - * associer à chaque mot clé une entité de l'ontologie
 - * construire l'interrogation ontologique
 - * interroger l'ontologie
 - * générer la réponse à donner à l'apprenant à partir de la réponse obtenue de l'ontologie

2.4 Architecture du système

Nous allons vous présenter ici l'architecture fonctionnelle de notre système et ensuite une architecture détaillée.

2.4.1 Architecture globale

Pour représenter l'architecture globale ou encore l'architecture fonctionnelle de notre système, nous avons choisi d'utiliser une architecture en couche qui se présente comme suite :

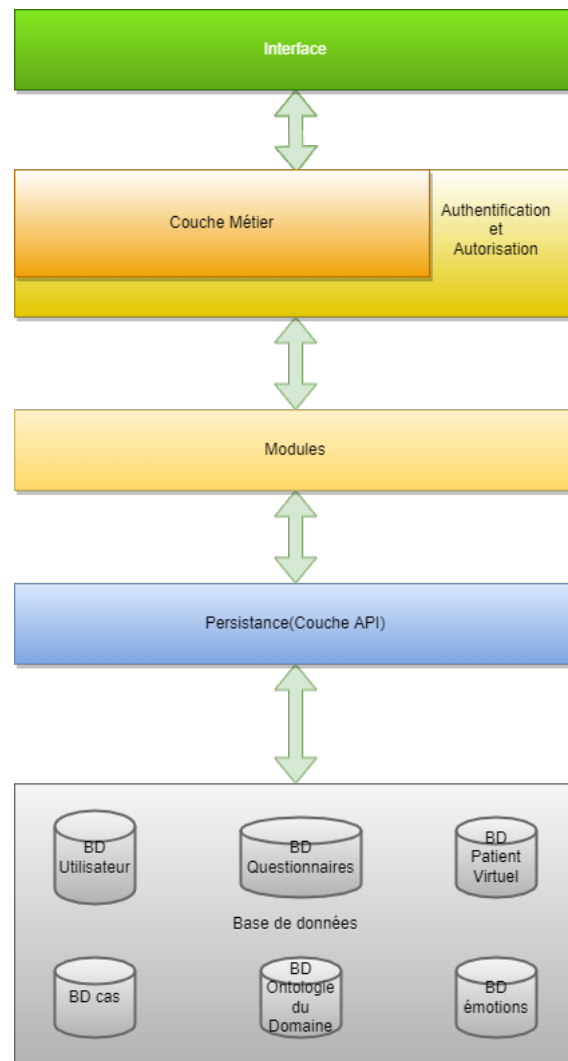


FIGURE 2.16. Architecture Globale

L'architecture en couches organise le système en couches avec des fonctionnalités connexes associées à chaque couche. Une couche fournit des services à la couche précédente, de sorte que les couches de plus bas niveau représentent les services de base susceptibles d'être utilisés dans tout le système.

Cette architecture est utilisée lors de la construction de nouvelles installations par dessus des systèmes existants ; lorsque le développement est réparti entre plusieurs équipes, chaque équipe étant responsable d'une couche de fonctionnalité.

2.4.2 Architecture détaillée

Ici nous allons présenter l'architecture détaillée de la couche modules mis en évidence au niveau de l'architecture globale du système. Cette couche représente l'ensemble des modules de notre système à savoir le module expert, le module tuteur et le module de l'apprenant.

Nous allons la présenter en mettant en exergue les différents algorithmes et notions mathématiques utilisés dans chaque module et ainsi que la communication entre les modules. Le schéma ci-dessous est le résultat de notre travail.

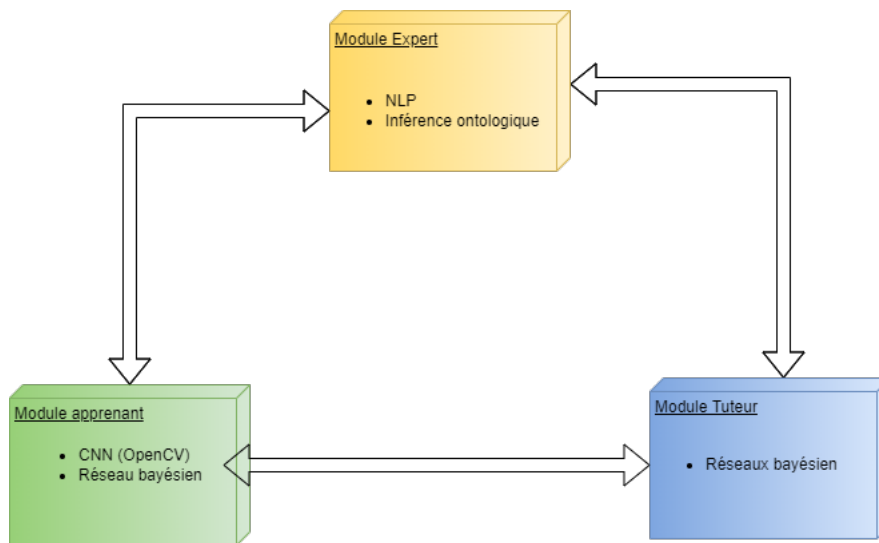


FIGURE 2.17. Architecture détaillée de la couche modules

2.5 Implémentation

2.5.1 Choix des outils

Pour mettre en oeuvre le système ainsi modélisé, nous avons utilisé les outils listés ci après :

2.5.1.1 Blender

Blender est un puissant outil de développement pour créer des images et animations 3D. C'est un logiciel libre et complètement gratuit. Il prend en charge l'intégralité du pipeline 3D : modélisation, rigging, animation, simulation, rendu, composition et suivi de mouvement, montage vidéo et création de jeux. Blender est **multiplateforme** et fonctionne aussi bien

sur les systèmes linux, windows et mac. Il est bien adapté aux particuliers, et aux petits studios qui bénéficient de son pipeline unifié et de son processus du développement réactif

* **Avantages**

- Blender est libre et open source, il peut donc être personnalisé par l'utilisateur avec des scripts Python.
- Il dispose d'une grande communauté d'utilisateur et d'une documentation accessible à tous, ainsi que des tutoriels.
- Les projets blender peuvent être exporté dans différents formats, parmi lesquels certains sont facilement intégrable sur le web.

* **Inconvénients**

- Blender étant puissant, consomme énormément de ressources machine tel que la RAM, processeur et la batterie.
- La prise en main de la modélisation 3D avec Blender n'est pas aisé, et nécessite beaucoup de temps de d'apprentissage et d'adaptation, surtout pour la gestion des mouvements et animations



FIGURE 2.18. Logo Blender

2.5.1.2 ReactJS

C'est une bibliothèque JavaScript pour la construction d'interfaces utilisateur (UI). Elle s'appuie sur des composants autonomes qui maintiennent leur propre état et que l'on peut assembler pour créer des interfaces complexes. Puisque l'interface utilisateur devra présenter un patient virtuel, il nous faudra, pour présenter l'environnement et le patient virtuel 3D créés avec Blender, utiliser les éléments suivants :

- * **script Blender** : c'est un code javascript décrivant l'environnement crée précédemment sur Blender

- * **ThreeJS** : c'est une bibliothèque JavaScript multi-navigateurs et une interface de programmation d'applications utilisée pour créer et afficher des images de synthèse 3D animées dans un navigateur Web à l'aide de WebGL.
- * **React Three Fiber** : c'est un puissant moteur de rendu Three.js qui permet de rendre des modèles et des animations 3D pour React et ses applications natives.

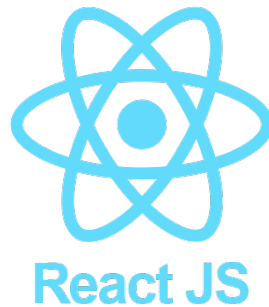


FIGURE 2.19. Logo ReactJS

2.5.1.3 DjangoREST Framework (DRF)

Django REST est un framework puissant et flexible avec une architecture modulaire et personnalisable qui vise à construire des API web sophistiquées. Ce framework utilise Python et Django.



FIGURE 2.20. Logo Django REST

Nous utiliserons Django Rest pour créer les interfaces applicatives de notre système en implémenter tous les modules cités précédemment. Pour ce faire, nous devons utiliser les outils tels que :

- * **PyTorch** : c'est une bibliothèque open source d'apprentissage automatique basée sur la bibliothèque **Torch**, utilisée pour des applications telles que la vision par ordinateur et le traitement du langage naturel.



FIGURE 2.21. Logo PyTorch

* **PyAgrum** pyAgrum est une bibliothèque scientifique C++ et Python dédiée aux réseaux bayésiens (BN) et autres modèles graphiques probabilistes. Basée sur la bibliothèque C++ aGrUM, elle fournit une interface de haut niveau à la partie C++ d'aGrUM permettant de créer, gérer et effectuer des calculs efficaces avec les réseaux bayésiens et autres modèles graphiques probabilistes.

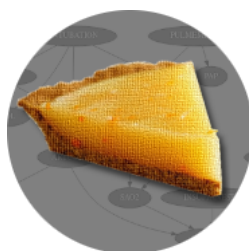


FIGURE 2.22. Logo PyAgrum

2.5.1.4 UMLS

UMLS (Unified Medical Language System) est Nous utiliserons l'API d'UMLS accessible gratuitement pour obtenir l'ontologie relative au diagnostic médical, ce qui nous permettra de construire la base de connaissance.

Description	Hiérarchies	Relations	PubMed / DocCISMeF
<div> <div>Intra-terminologiques</div> <div>Inter-terminologiques</div> </div>			
affecte (30)			
est affecté(e) par (45)			
effet évalué par (1)			
associé avec (24)			
Alignements automatiques exacts (par équipe CISMeF) (4)			
causé(e) par (36)			
co-occure avec (10)			
complique (10)			
est compliqué(e) par (18)			
conceptuellement relié(e) à (1)			
a pour degré(s) (6)			

FIGURE 2.23. Aperçu de la terminologie "Maladie" décrite par UMLS

2.5.1.5 OntoDB(2) (Ontologic DataBase)

Les bases de données à base ontologique sont des bases de données qui est les bases de données qui contiennent à la fois des données et des ontologies qui en décrivent la sémantique. L'interrogation de ces bases de données peut se faire des langages d'exploitation des bases de données à base ontologique telle que **OntoQL**.

2.5.1.6 PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle libre et gratuit qui met l'accent sur l'extensibilité et la conformité SQL. Il présente les avantages d'être compatible ACID et traiter les données dans un délai raisonnable. En plus de cela, PostgreSQL est non seulement compatible avec **gizmos**, un utilitaire python pour le développement des ontologies mais est aussi un prérequis pour l'utilisation d'OntoDB, une base de données à base ontologique.



FIGURE 2.24. Logo PostgreSQL

2.5.1.7 OntoNLQA

Ontology Natural Language Question Answering est un framework basé sur le traitement automatique du langage qui permet d'interroger une ontologie à partir du langage naturel. Ce framework interroge une ontologie en utilisant SPARQL.

Outils	Utilisation
Blender	Modélisation 3D du patient virtuel et son environnement
ReactJS	Interface utilisateur
Django REST	Interface applicative
PyTorch	Traitement du langage naturel (NLP)
PyAgrum	Réprésentation des réseaux bayesiens pour la modélisation des connaissances
OpenCV	Reconnaissances des émotions de l'apprenant
UMLS API	Acquérir les ontologies du domaine médical
PostgreSQL	Système de gestion de base de données (cas, apprenant, questionnaire, émotions, etc...)
OntoDB	Réprésenter les ontologies sous forme de base de données à base ontologique
OntoNLQA	Interroger les ontologies à partir du langage naturel
GPT-3	Générer la réponse à afficher à l'apprenant fonction de la réponse obtenue de l'ontologie

TABLE 2.2 – Choix des outils d'implémentation

2.6 Déploiement

Cette partie tient lieu de guide pour effectuer le déploiement du travail réalisé. Le diagramme de déploiement permet de décrire l'architecture de déploiement de

notre solution.

Ici nous distinguons 4 composants principaux à savoir :

- Le terminal (l'ordinateur)
- Le serveur web
- Le serveur d'application
- Le serveur de base de données

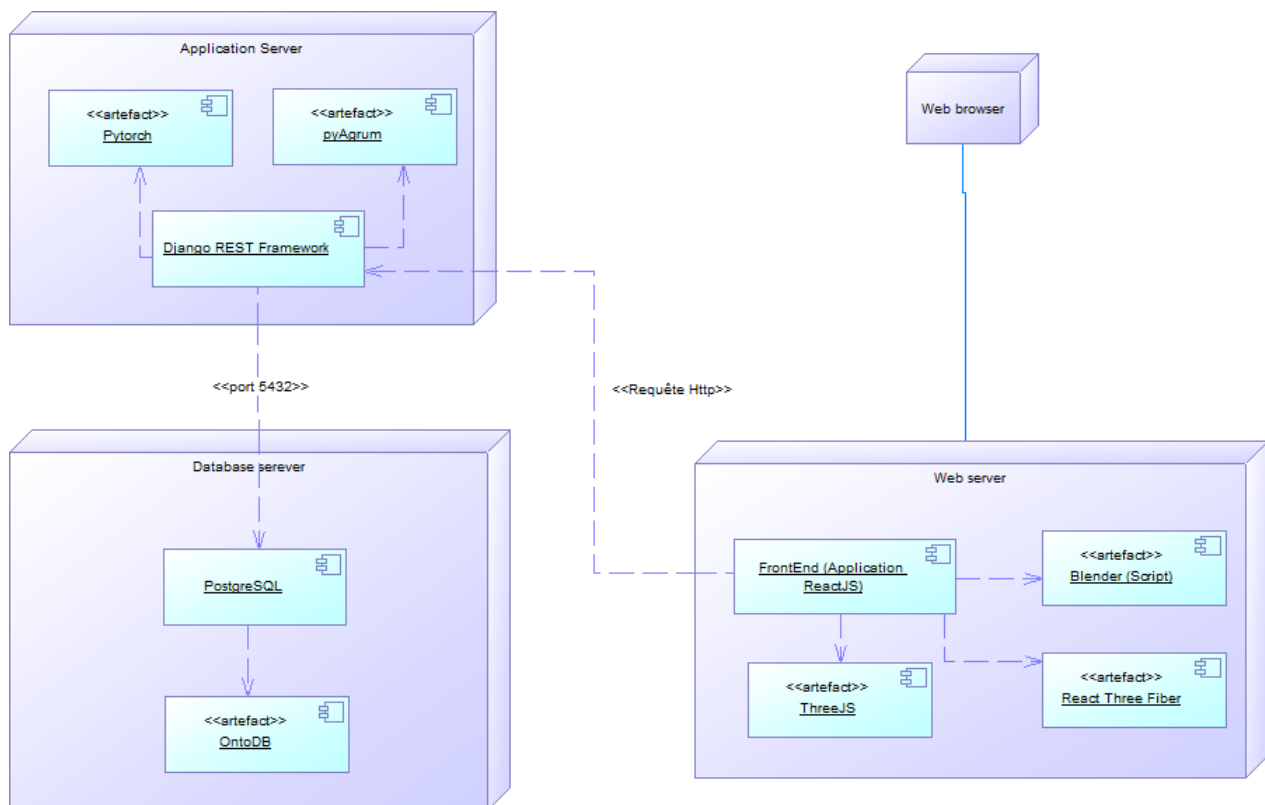


FIGURE 2.25. Diagramme de déploiement

EXPÉRIMENTATIONS ET RÉSULTATS

Dans cette partie il est question pour nous d'évaluer ce que nous avons fait en émettant des hypothèses et montrer comment ces dernières seront validées en mettant en exergue les résultats obtenus.

Sommaire

3.1	Rappel du problème	49
3.2	Le module de l'expert	49
3.3	Le module du tuteur	49
3.4	Le module de l'apprenant	49

3.1 Rappel du problème

Il était question pour nous de concevoir un système tutoriel intelligent pour l'aide à la pose diagnostic.

Dans cette partie, il est question pour nous de faire des hypothèses et de les valider. Pour le faire, nous allons procéder par établissement des hypothèses et validation pour chacun de nos modules parmi lesquelles : le module de l'expert, le module de tuteur, le module de l'apprenant et l'interface utilisateur.

3.2 Le module de l'expert

Supposons un médecin expert qui souhaite entrer des connaissances pour un ou plusieurs cas de maladies.

Pour ce faire, il suffit que le médecin expert se connecte au système et suis la procédure pour entrer les cas de maladies qui seront sauvegardées dans la base de connaissances.

Si le médecin expert souhaite modifier ou supprimer les connaissances qui sont contenues dans ces bases de connaissances, alors il suffit qu'il suit la procédure associée pour effectuer l'opération souhaitée.

3.3 Le module du tuteur

Supposons un apprenant qui souhaite suivre une formation dans un domaine de la médecine.

Pour ce faire, il suffit que l'apprenant choisisse le domaine dans lequel il aimerait se former et le modèle du tuteur va se charger de d'établir le plan et le temps approprié pour l'acquisition des connaissances et l'afficher à l'apprenant.

3.4 Le module de l'apprenant

Supposons que l'apprenant veut faire une évaluation formative.

Il aura à choisir le cas sur lequel il veut travailler et commencer la consultation avec le patient virtuel. Il pourra enregistrer les évidences qui proviennent des questions qu'il a posées au patient et à l'issu de cela, il peut poser des hypothèses.

Lorsqu'il émet des hypothèses, il continue ses investigations en proposant des examens pour affiner les hypothèses faites. Si lors de ses investigations, il pose des actes qui laissent

croire au tuteur que l'apprenant a changé complètement d'hypothèses, le tuteur intervient en lui demandant pourquoi il a changé d'hypothèses, quels sont les arguments qui lui permettent de changer d'hypothèses. et si l'hypothèse qu'il choisit est la voie correcte (celle entrée préalablement par l'expert), le tuteur le laisse continuer, sinon, il s'arrête et recommence à poser des hypothèses ainsi de suite jusqu'à la fin de la consultation. Cette rétroaction du tuteur est possible grâce au réseau bayésien développé au niveau du module du tuteur qui permettra au tuteur de suivre le travail de l'apprenant.

Si l'apprenant veut passer une évaluation sommative, il la passe normalement comme un contrôle, sans intervention du tuteur au cours de l'évaluation. A la fin de l'évaluation, il obtient son score global, et une indication sur son efficacité ; son niveau est mis à jour en fonction de sa performance lors de l'évaluation par le module expert.

DISCUSSION

Dans cette partie, il est question pour nous de faire une discussion de notre travail, c'est-à-dire énoncer les différents avantages et limites de notre solution.

Sommaire

4.1	Avantages	52
4.2	Limites	52

4.1 Avantages

Comme avantages de notre système nous pouvons citer :

- Une assez bonne représentation du patient virtuel ;
- Représentation du patient virtuel dans un environnement médical(salle de consultation) ;
- Utilisation d'un SGBD relationnel qui prend en compte le contexte des ontologies et sur laquelle on peut faire directement des inférences ontologiques ;
- Représentation des émotions du patient virtuel ;
- Utilisation des réseaux bayésiens pour la représentation de l'état de connaissance de l'apprenant ;
- L'apprenant peut consulter physiquement le patient dans ce sens il peut toucher le patient grâce à certains outils présenter à l'interface ;
- Dispose à la fois d'une évaluation sommative et formative.

4.2 Limites

Les limites sont :

- Le NLP utilisé pour comprendre la question et trouver la réponse peut fausser les résultats ;
- L'identification des émotions de l'apprenant peut ne pas être possible.

CONCLUSION GÉNÉRALE

RAPPEL DU PROBLEME

Il était question pour nous de proposer une modélisation d'un patient virtuel interactif pour la formation des jeunes médecins à la pose du diagnostic médical. Le but de ce système est d'améliorer la formation des jeunes médecins en leur fournissant un environnement au travers duquel ils pourront s'exercer à poser un diagnostic sur des cas de patients basés sur la réalité. Ceci permettra de :

- réduire le cas de faux diagnostic
- apporter une assistance professionnelle aux jeunes médecins

DEMARCHE ET RESULTAT

Pour y arriver, nous avons procédé comme suit :

- **Etat de l'art** : à cette étape, nous avons fait un benchmarking sur les différents patients virtuels existants basés sur systèmes tutoriels intelligents. Nous avons recherché comment fonctionne chacun des modules à savoir interface, apprenant, tuteur, expert ont été conçus enfin de répondre à la problématique. Il en ressort que :
 - * les ontologies sont adéquates pour la représentation des connaissances du domaine médical mais qu'il est préférable d'utiliser des bases de données à base ontologique enfin de gérer aussi la persistance des données
 - * les réseaux bayésiens quant eux présentent de bons résultats pour la représentation des connaissances de l'apprenant
- **Analyse et conception du système** : Sur la base des résultats obtenues de l'état de l'art et des objectifs qui nous ont été assignés, nous avons pu ressortir clairement les exigences de ce système et en découler un modèle conceptuel.
- **Modélisation du système** : Il était question lors de cette étape de définir de manière explicite comment chaque module devrait fonctionner. Pour cela, nous avons utilisé les concepts théoriques telles que : le traitement du langage naturel (pour comprendre et répondre aux questions posées par l'apprenant), la reconnaissance des émotions (pour déterminer).

- **Choix des outils d'implémentation** : Après avoir conçu un modèle du système, il était question de proposer des outils à l'aide desquels, il serait possible d'implémenter le système. Parmi ces outils nous pouvons citer : **PyAgrum** pour l'exploitation des réseaux bayesiens et **OntoDB** pour la gestion des bases de données à base ontologique.
- **Expérimentation et résultats** : A cette étape, nous avons émis des hypothèses qui permettrait de déterminer la véracité et les performances de notre modèle. De cette expérimentation, il en ressort que :
- **Discussion** : Grâce au résultat obtenu de l'expérimentation, nous avons pu ressortir les avantages et limites de notre modèle.

DIFFICULTES RENCONTREES

Au cours de la réalisation de ce travail, nous avons rencontrées plusieurs difficultés liées à :

- **L'accès au ressources** : nous aurions aimé lors de nos recherches avoir accès à tous les outils et à l'intégralité des articles mais la plupart de ces accès était conditionné par l'appartenance à une institution.
- **La compréhension de certaines notions**

PERSPECTIVES

- Le diagnostic orale entre le patient virtuel et l'apprenant
- Le patient virtuel doit être capable de faire des mouvements
- L'apprenant doit pouvoir choisir ses exercices

BIBLIOGRAPHIE

- [1] A. ALKHATLAN, K. KALITA. Intelligent Tutoring Systems : A Comprehensive Historical Survey with Recent Developments. Dec 2018.
- [2] H. Asiaee, T. Minning, L. Tarleton. A framework for ontology-based questionanswering with application to parasiteimmunology. Journal of Biomedical Semantics. Article 31. 2015
- [3] Mémoire Monthe Valery
- [4] Un tuteur intelligent pour les activités du raisonnement clinique. 2020
- [5] Bayesian Networks in Medicine : a Model-based Approach to Medical Decision Making of Peter Lucas Department of Computing Science University of Aberdeen Scotland, UK plucas@csd.abdn.ac.uk
- [6] <https://www.iro.umontreal.ca/~frasson/cours/dift6243/6243-4-Systemes-\tutoriels-int.pdf>, consulté le 13 décembre 2021 à 17h30
- [7] https://www.hetop.eu/hetop/fr/#rr=UML_ST_T047&oti=T_DESC_UMLS_SEMANTIC_TYPE,T_DESC_UMLS_SEMANTIC_GROUP&q=maladie, consulté le 07 décembre 2021 à 16h14
- [8] <https://www.nlm.nih.gov/hsrph.html>, consulté le 07 décembre 2021 à 16h20
- [9] <https://news.un.org/fr/story/2019/09/1051752>, consulté le 08 décembre 2021 à 15h00
- [10] https://wiki.telug.ca/wikitedia/index.php/Syst%C3%A8me_tutoriel_intelligent, consulté le 15 janvier 2022 à 17h50
- [11] <https://www.snomed.org/>, consulté le 10 janvier 2022 à 13h00
- [12] <https://www.academie-medecine.fr/06-12-le-diagnostic-en-medecine-histoire-mise-en-uvre-presente-perspectives/>, consulté le 12 janvier 2022 à 10h00
- [13] <https://www.deuxiemeavis.fr/blog/article/310-le-diagnostic-medical-les-etapes-pour-trouver-votre-maladie>, consulté le 09 janvier 2022 à 10h00
- [14] <http://ihtsdo.github.io/sct-snapshot-rest-api/api.html>, consulté le 13 janvier 2022 à 15h00

- [15] <https://github.com/IHTSDO/sct-snapshot-rest-api>, consulté le 12 janvier 2022 à 3h00
- [16] <https://snomedctsnapshotapi.docs.apiary.io/#reference/\releases/release-information/releases>, consulté le 10 janvier 2022 à 17h20
- [17] <https://pyagrum.readthedocs.io/en/0.22.5/>, consulté le 014 janvier 2022 à 18h00
- [18] https://forge.lias-lab.fr/projects/ontodb/wiki/Installation_v1, consulté le 13 janvier 2022 à 15h30
- [19] https://www.cmpa-acpm.ca/serve/docs/ela/goodpracticesguide/pages/manage_risk/The_diagnostic_process/differential_diagnosis-f.html, consulté le 15 janvier 2022 à 18h00
- [20] <http://europepmc.org/article/med/26185615>, consulté le 19 janvier 2022 à 19h00
- [21] <https://github.com/HHS/uts-rest-api>, consulté le 17 janvier 2022 à 19h00
- [22] <https://tel.archives-ouvertes.fr/tel-00201777>, consulté le 18 janvier 2022 à 15h00
- [23] <https://documentation.uts.nlm.nih.gov/rest/home.html>, consulté le 08 janvier 2022 à 10h00
- [24] <https://www.nlm.nih.gov/research/umls/quickstart.html>, consulté le 19 décembre 2021
- [25] Patients virtuels : pédagogie, état de l'art et développement du simulateur Alphadiag