



**UNIVERSITÉ «ABBES LAGHROUR » DE KHENCHELA**  
FACULTÉ DES SCIENCES ET LA TECHNOLOGIES  
DÉPARTEMENT DE MATHÉMATIQUE ET INFORMATIQUE



## **Mémoire de fin d'études**

Pour l'obtention du diplôme Master en Informatique (L.M.D.)

**Option : Sécurité et Technologies Web**

# **Conception et réalisation d'un système d'aide à la prise de décision d'un diagnostic médical.**

*Réalisé par :*

Mlle. KHELALFA Malak

Mlle. ALLOUCHE Sara Malek

*Encadré par :*

Dr. CHERGUI Wahid

*Soutenu le 21 Juin 2023, Devant le jury composé de :*

Dr. Hemmam Mounir :

Dr. Chergui Wahid :

Mr. Ben Athmane Mohammed:

Président

Rapporteur

Examineur

# Dédicace

“

*À moi et À mes êtres chers, piliers de mon projet de fin d'études, votre soutien inconditionnel, amour et encouragement ont été essentiels.*

*Merci pour cette aventure passionnante,*

*À Ma famille, mon rocher et ma source d'inspiration, votre amour, confiance et soutien indéfectibles ont réalisé mes rêves. Vos encouragements constants ont été ma force motrice. Je suis honorée d'avoir une famille aussi aimante et solidaire,*

*À Mes amis proches, compagnons d'aventure, votre présence, vos encouragements et discussions stimulantes ont rendu ce parcours mémorable. Chaque moment partagé avec vous a été une source de motivation. Je vous suis reconnaissante d'être à mes côtés,*

*À mes enseignants dévoués, votre expertise et passion ont été essentiels à ma croissance académique en intelligence artificielle. Vos encouragements et conseils avisés ont fait de moi un apprenant accompli et je vous en suis profondément reconnaissante,*

*Je dédie ce travail à mon binôme Sara, pour leur précieuse collaboration et contribution tout au long de ce projet. Merci pour cette expérience enrichissante.*

*je vous aime*

”

**- Malek**

# Dédicace

“

*À mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse et leur soutien tout au long de mes études .*

*À toi, ma chère tante, ma deuxième mère, je te remercie du fond du cœur pour ton amour inconditionnel, ta présence constante et ta bienveillance infinie. Ce mémoire est dédié à toi, en reconnaissance de tout ce que tu as fait pour moi.*

*À mes sœurs et ma grande mère, je vous remercie du fond du cœur pour votre amour inconditionnel, votre soutien indéfectible et vos encouragements.*

*À mes chers amis, notre amitié précieuse a été une source constante de soutien, de joie et d'inspiration tout au long de cette aventure académique.*

*À mes professeurs, Votre savoir et votre passion m'ont inspiré. Merci pour votre dévouement inébranlable, ce mémoire est le fruit de votre enseignement. Je vous suis reconnaissante à jamais*

*À mon binôme Malek, je souhaite dédier ce mémoire à toi, ta présence, ton soutien et ta collaboration ont été essentiels à notre réussite commune.*

”

- Sara Malek

# Remerciements

Nous souhaitons exprimer notre reconnaissance à "Allah" qui nous a guidés tout au long de ce travail, nous inspirant à faire les bons choix et à adopter les bonnes attitudes. Sans Sa miséricorde, ce projet n'aurait jamais été réalisé.

Ce mémoire de fin d'études a été accompli à l'université **ABBES LAGHROUR** à **Khenchela**. Nous tenons à exprimer notre profonde gratitude envers toutes les personnes qui nous ont soutenus et ont contribué à l'élaboration de ce mémoire, ainsi qu'à la réussite de cette année universitaire exceptionnelle.

Nous souhaitons exprimer notre sincère gratitude à Docteur **CHERGUI WAHID**, notre encadrant, qui a été constamment présent et disponible tout au long de la réalisation de ce mémoire. Nous lui sommes reconnaissants pour son inspiration, son soutien et le temps précieux qu'il nous a accordé. Sans lui, ce mémoire n'aurait jamais pu être réalisé. "Nous vous remercions pour vos conseils, votre persévérance et votre bienveillance".

Nous voulons également exprimer notre profonde gratitude envers l'ensemble du corps professoral et administratif de la Faculté des Sciences, département de mathématiques et informatique.

Nos remerciements vont également aux membres du jury pour avoir accepté d'évaluer ce travail et pour leurs suggestions précieuses.

Nous adressons une pensée spéciale à toutes les personnes que nous avons croisées sur notre chemin, que ce soit à l'université ou ailleurs dans ce vaste monde.

# Résumé

Dans notre étude, nous avons choisi de nous concentrer sur le diagnostic du diabète dans sa globalité, sans faire de distinction entre les différents types de diabète. Cette approche vise à aborder le diabète en tant que maladie complexe, nécessitant une évaluation minutieuse des symptômes, des antécédents médicaux et des résultats de tests pour parvenir à un diagnostic précis. En se focalisant sur le diabète dans son ensemble, nous souhaitons développer un système d'aide à la décision qui puisse être applicable à tous les patients suspects de diabète.

Cette approche permettra d'obtenir un diagnostic basé sur des preuves solides en analysant les similitudes entre les cas et en les comparant aux données d'évaluation du patient. En utilisant des techniques de raisonnement à partir des cas et en exploitant les avancées de l'intelligence artificielle et de l'apprentissage automatique, notre système pourra continuellement s'améliorer le diagnostic du diabète.

L'objectif ultime est de faciliter le processus de diagnostic du diabète, d'améliorer la précision des résultats et de permettre une prise en charge précoce et efficace des patients atteints de cette maladie chronique.

---

**Mots clés :** le diagnostic du diabète, Système d'Aide, , Prise de décision, L'apprentissage automatique, Intelligence Artificielle, Raisonnement à partir de cas.

---

# Abstract

In our study, we have chosen to focus on the diagnosis of diabetes as a whole, without distinguishing between different types of diabetes. This approach aims to address diabetes as a complex disease, requiring a thorough evaluation of symptoms, medical history, and test results to achieve an accurate diagnosis. By focusing on diabetes as a whole, we aim to develop a decision support system that can be applicable to all patients suspected of having diabetes.

This approach will enable us to obtain a diagnosis based on solid evidence by analyzing similarities between cases and comparing them to the patient's evaluation data. By using case-based reasoning techniques and leveraging advancements in artificial intelligence and machine learning, our system will continuously improve and refine its the diagnosis of diabetes.

Our ultimate goal is to facilitate the process of diagnosing diabetes, improve the accuracy of the results, and enable early and effective management of patients with this chronic disease.

---

**Keywords :** The diagnosis of diabetes, System of Assistance, Make decisions, decision support system , Artificial Intelligence, machine learning, Case-Based Reasoning.

---

## ملخص

في دراستنا، اخترنا التركيز على تشخيص مرض السكري بشكل شامل، دون التمييز بين مختلف أنواعه. تهدف هذه الدراسة إلى التعامل مع مرض السكري كمرض معقد، يتطلب تقييمًا دقيقًا للأعراض والسجلات الطبية ونتائج الاختبارات للوصول إلى تشخيص دقيق. من خلال التركيز على مرض السكري بشكل عام، نسعى لتطوير نظام دعم القرار الذي يمكن تطبيقه على جميع المرضى المشتبه في إصابتهم بالسكري، بغض النظر عن نوع السكري الذي يعانون منه.

ستسمح هذه النهج بتقديم تشخيصات مستندة إلى أدلة قوية من خلال تحليل أوجه التشابه بين الحالات ومقارنتها ببيانات المريض الجاري تقييمه. باستخدام تقنيات الاستدلال من الحالات واستغلال التطورات في مجال الذكاء الاصطناعي والتعلم الآلي، سيتمكن نظامنا من التحسين المستمر في عملية تشخيص مرض السكري.

الهدف النهائي هو تسهيل عملية تشخيص مرض السكري وتحسين دقة النتائج وتمكين التدخل المبكر والفعال للمرضى الذين يعانون من هذا المرض المزمن. ومقارنتها ببيانات تقييم المريض

---

**الكلمات المفتاحية:** نظام المساعدة، صنع القرار، الذكاء الاصطناعي، تشخيص مرض السكري، التعلم الآلي، الاستدلال على أساس الحالات.

---

# Table des matières

Dédicace . . . . .	I
Dédicace . . . . .	II
Remerciements . . . . .	III
Résumé . . . . .	IV
Abstract . . . . .	V
VI . . . . .	ملخص
Introduction générale . . . . .	1
<b>1 Les système d'aide à la décision médicale . . . . .</b>	<b>4</b>
1.1 Introduction . . . . .	5
1.2 La prise de décision . . . . .	5
1.2.1 Le processus de la prise de décision . . . . .	5
1.2.2 Les phases du processus de prise de décision . . . . .	6
1.3 Les systèmes d'aide à la décision . . . . .	7
1.3.1 L'utilisation des SAD . . . . .	7
1.3.2 Classification des SAD : . . . . .	7
1.3.3 La combinaison de l'IA et des systèmes d'aide à la décision . . . . .	8
1.3.4 La décision médicale . . . . .	9
1.4 Le système aide à la décision médicale . . . . .	10
1.4.1 Définition . . . . .	10
1.4.2 La typologie des SADM . . . . .	10
1.4.3 Structure d'un SADM . . . . .	11
1.4.4 Les principales fonctions du SADM . . . . .	11
1.4.5 Les objectifs du SADM . . . . .	12
1.5 Conclusion . . . . .	13
<b>2 Raisonnement à partir de cas . . . . .</b>	<b>14</b>
2.1 Introduction . . . . .	15
2.2 Bref historique . . . . .	15
2.3 Raisonnement à partir de cas . . . . .	15
2.4 Connaissance de système à partir de cas . . . . .	16
2.5 Description générale du cas . . . . .	17
2.5.1 Définition du cas . . . . .	17

2.5.2	Représentation des cas . . . . .	18
2.5.3	Indexation des cas . . . . .	18
2.5.4	Stokage des cas . . . . .	19
2.6	Le cycle de RàPC . . . . .	19
2.6.1	Phase de la Récupération des cas similaires(RETRIEVE) : . . . . .	20
2.6.2	Phase de l'adaptation des cas trouvés(REUSE) : . . . . .	22
2.6.3	Phase de la Révision des solutions(REVISE ) : . . . . .	23
2.6.4	Phase de rétention (RETAIN) : . . . . .	23
2.7	Quelques projets basés sur le RàPC . . . . .	23
2.8	Domaines d'application . . . . .	24
2.9	Les avantages du raisonnement à partir de cas . . . . .	24
2.10	Conclusion . . . . .	25
<b>3</b>	<b>L'Application du RàPC dans un système d'aide à la décision pour le diagnostic du Diabète . . . . .</b>	<b>26</b>
3.1	Introduction . . . . .	27
3.2	Les types des systèmes RàPC en médecine . . . . .	27
3.2.1	Les caractéristiques orientées vers les objectifs . . . . .	27
3.2.2	Les caractéristiques orientées vers sa construction . . . . .	28
3.3	Les limites de l'aide à la décision médicale par RàPC . . . . .	28
3.4	Exemples de SADM fondés sur un RàPC . . . . .	29
3.5	Le RàPC comme un système d'aide à la décision pour le diagnostic du Diabète . . . . .	30
3.5.1	La Définition du diabète[10] . . . . .	30
3.5.2	Les Types du diabète . . . . .	30
3.5.3	Symptômes du diabète . . . . .	31
3.5.4	Diagnostic du diabète . . . . .	31
3.5.5	Le raisonnement à partir de cas pour le diagnostic du Diabète . . . . .	32
3.6	Conclusion . . . . .	32
<b>4</b>	<b>Le framework JColibri . . . . .</b>	<b>33</b>
4.1	Introduction . . . . .	34
4.2	COLIBRI Studio[28] . . . . .	34
4.3	Les assistants . . . . .	34
4.3.1	New CBR project wizard . . . . .	35
4.3.2	Wizard New k-NN CBR Project . . . . .	35
4.3.3	New CBR system from template . . . . .	36
4.4	Les Outils principaux de COLIBRI studio . . . . .	36
4.4.1	L'outil Case Designer . . . . .	36
4.4.2	L'outil similarity . . . . .	36
4.4.3	L'outil Case Base Selector . . . . .	36
4.4.4	L'outil plain text connector configurator . . . . .	37
4.5	jCOLIBRI Framework . . . . .	38
4.5.1	jCOLIBRI version 3 . . . . .	38
4.6	Architecture de jCOLIBRI3 . . . . .	38
4.7	Les interfaces principales de jCOLIBRI[29] . . . . .	39

4.7.1	Interface StandardCBRAApplication . . . . .	39
4.7.2	Interface CaseComponent . . . . .	39
4.7.3	Interface CBRCaseBase . . . . .	40
4.7.4	Interface Connector . . . . .	40
4.8	Les fonctions de similarité . . . . .	42
4.9	Conclusion . . . . .	43
<b>5</b>	<b>Conception . . . . .</b>	<b>44</b>
5.1	Introduction . . . . .	45
5.2	Etude de cas : diagnostic de diabète . . . . .	45
5.3	Présentation détaillée de notre Base de connaissance : . . . . .	45
5.4	Modélisation de la solution : . . . . .	46
5.4.1	La représentation des données : . . . . .	46
5.4.2	Gestion des connecteurs : . . . . .	47
5.4.3	Configurations des structures de cas et des similarités.: . . . . .	47
5.4.4	Creation de CBR application : . . . . .	49
5.4.5	Vue globale de notre système : . . . . .	51
5.5	Conclusion . . . . .	57
<b>6</b>	<b>Implementation . . . . .</b>	<b>58</b>
6.1	Introduction . . . . .	59
6.2	Technologies utilisées . . . . .	59
6.3	Installation de COLIBRI dans Eclipse . . . . .	61
6.4	Présentation du notre système . . . . .	65
6.5	Aperçu visuel de notre système . . . . .	66
6.5.1	Accueil . . . . .	66
6.5.2	Interface Nouveau cas . . . . .	67
6.5.3	Interface Cas retrouvés . . . . .	68
6.5.4	Interface Retenir . . . . .	70
6.6	Test et évaluation du système . . . . .	74
6.6.1	Évaluation et test de la performance du notre système CBRDD . . . . .	74
6.7	Conclusion . . . . .	76
	<b>Conclusion générale . . . . .</b>	<b>77</b>

# Table des figures

1.1	Les phases du processus de prise de décision[3]. . . . .	6
1.2	Techniques intelligentes en aide à la décision[1] . . . . .	9
2.1	Conteneur de connaissance. . . . .	17
2.2	Le cycle du RàPC . . . . .	20
4.1	Assistants de COLIBRI studio . . . . .	35
4.2	Capture d'écran de l'outil Case Selector . . . . .	37
4.3	Architecture de jCOLIBRI 3. [8] . . . . .	39
4.4	le diagramme UML de jCOLIBRI.[2] . . . . .	41
5.1	représentation des cas . . . . .	47
5.2	Configurations des structures de cas et des similarités . . . . .	48
5.3	la conception globale de notre système. . . . .	49
5.4	le code de représentation de CasDescription. . . . .	49
5.5	le code de représentation de casSolution . . . . .	50
5.6	Plain Text Connector Config. . . . .	50
5.7	la conception globale de notre système. . . . .	51
5.8	Code java pour l'ajout de nouveau cas. . . . .	52
5.9	Code java " Récupération de cas similaires". . . . .	53
5.10	Code java " les fonctions de similarity". . . . .	53
5.11	Code java " l'étape d'adaptation". . . . .	54
5.12	Code java " l'etapde de révision et retention". . . . .	55
5.13	Code java " l'enregistrement dans Data temporaire". . . . .	56
5.14	Digramme de classe. . . . .	57
6.1	Logo de Eclipse. . . . .	59
6.2	Logo de Colibri studio. . . . .	59
6.3	Logo de Jcolibri3. . . . .	60
6.4	Logo de MySQL Workbench. . . . .	60
6.5	Logo de langage Java. . . . .	60
6.6	Logo de kaggle. . . . .	61
6.7	Logo UML. . . . .	61
6.8	Étape 1 :Selection de l'installation du logiciel. . . . .	62
6.9	Étape 2 :Importation de COLIBRI à partir de l'installation existante de l'application . . . . .	63
6.10	Étape 3 : Vérification des éléments à installer". . . . .	63
6.11	"Étape 4 : Vérification des licences". . . . .	64
6.12	Ouverture de la perspective COLIBRI. . . . .	65

6.13 KNN projet . . . . .	65
6.14 Interface accueil. . . . .	67
6.15 Interface Nouveau Cas. . . . .	68
6.16 Interface Cas retrouvés. . . . .	69
6.17 Exemple de nouveau cas et les cas retrouvés . . . . .	70
6.18 Interface retenir. . . . .	70
6.19 Le premier Cas sélectionné . . . . .	71
6.20 Le nouveau cas validé et identifier par son nouveau ID . . . . .	72
6.21 Enregistrement de nouveau cas dans la base temporaire. . . . .	73
6.22 Enregistrement de data temporaire en MySQL . . . . .	73
6.23 L'affichage de data temporaire dans le tableau . . . . .	74

# Liste des tableaux

1.1	Fonction d'aide à la décision et exemple de problèmes cliniques[22] . . . . .	12
6.1	Résultats de l'évaluation de notre système CBRDD . . . . .	75

# Liste des sigles et acronymes

<b>SAD</b>	<i>Système d'aide à la décision</i>
<b>AI</b>	<i>Artificial Intelligence</i>
<b>SADM</b>	<i>système aide à la décision médicale</i>
<b>RàPC</b>	<i>Raisonnement à partir de cas</i>
<b>CBR</b>	<i>Case Based Reasoning</i>
<b>XML</b>	<i>eXtensible Markup Language</i>
<b>KPPV</b>	<i>K plus proches voisin</i>
<b>HGPO</b>	<i>Hyperglycemic-Induced Glucose Challenge</i>
<b>AVC</b>	<i>Accident vasculaire cérébral</i>
<b>HbA1c</b>	<i>Hemoglobin A1c</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>DBA</b>	<i>Data Base administrator</i>
<b>SQL</b>	<i>Structured Query language</i>
<b>JVM</b>	<i>Java virtual machine</i>
<b>UML</b>	<i>Unified Modeling Language.</i>
<b>CBRDD</b>	<i>Case Base Reasoning Diabetes Diagnostic</i>

# Introduction générale

## Contexte

Dans un monde où les maladies chroniques telles que le diabète connaissent une prévalence croissante, il devient primordial d'améliorer les outils de diagnostic pour permettre une prise en charge précoce et efficace des patients. C'est dans ce contexte que s'inscrit notre projet de développement d'un système d'aide à la décision basé sur le raisonnement à partir des cas pour le diagnostic du diabète.

Étant donné que le diabète est une maladie complexe, une évaluation minutieuse des symptômes et des résultats des tests est essentielle pour parvenir à un diagnostic précis. Cependant, les professionnels de la santé font face à de nombreux défis lorsqu'il s'agit d'analyser ces données et de prendre des décisions éclairées. C'est là que notre système d'aide à la décision entre en jeu.

C'est pour ça en utilisant une approche basée sur le raisonnement à partir des cas, donc notre système s'appuie sur une base de cas existante contenant des informations médicales détaillées sur des patients atteints de diabète ou En analysant les similitudes entre les cas et en les comparant aux données du patient en cours d'évaluation, notre système peut fournir des recommandations de diagnostic fondées sur des preuves solides. C'est pourquoi l'utilisation d'une approche de raisonnement par cas dans notre système est basée sur une règle de cas existante contenant des informations médicales détaillées sur les patients atteints de diabète ou non.

## Problématique

le diagnostic du diabète est une tâche complexe qui peut être compliquée par la variété des symptômes, des antécédents médicaux et des résultats de tests. Donc comment développer un système d'aide à la décision basé sur le raisonnement à partir des cas qui permette aux professionnels de la santé de mieux gérer ces informations complexes et de prendre des décisions éclairées pour un diagnostic précis du diabète ?

# Objectifs

L'objectif de ce projet est donc de proposer un système d'aide à la décision basé sur le raisonnement à partir des cas pour le diagnostic du diabète. Ce système permettra aux professionnels de la santé d'accéder à une base de connaissances étendue et d'obtenir des décisions fiables et précises pour le diagnostic du diabète. En utilisant les cas déjà connus, le système pourra bénéficier de l'expérience accumulée et améliorer la précision des diagnostics.

Ce projet vise à combiner les avancées en intelligence artificielle, en apprentissage automatique et en médecine pour améliorer la prise de décision médicale dans le domaine du diagnostic du diabète. En développant un système efficace, il pourrait contribuer à un dépistage plus précoce et à une meilleure gestion de cette maladie chronique, entraînant ainsi des bénéfices importants pour les patients et les professionnels de la santé.

# Organisation du mémoire

Ce mémoire est organisé en six chapitres :

Le premier chapitre "**Les système d'aide à la décision médicale**" Dans ce chapitre, nous étudions le processus de prise de décision en examinant ses différentes étapes. Nous explorons également comment les systèmes d'aide à la décision sont combinés avec l'intelligence artificielle pour améliorer leur efficacité. De plus, nous nous concentrons spécifiquement sur le système d'aide à la décision médicale, en examinant comment il est utilisé dans le domaine de la santé pour soutenir les professionnels de la santé dans leurs décisions médicales.

Le deuxième chapitre "**Le raisonnement à partir de cas**" Ce chapitre nous offre l'opportunité d'approfondir notre exploration des principes fondamentaux du Raisonnement à partir de cas (RaPC). Nous examinerons également quelques projets qui appliquent cette approche de raisonnement, ainsi que les avantages et limites associés. En somme, ce chapitre vise à nous fournir une compréhension approfondie des bases théoriques du RaPC, à saisir ses avantages et à nous offrir une vue d'ensemble de son utilisation dans le domaine de l'Intelligence Artificielle.

Le troisième chapitre "**L'Application du RàPC dans un système d'aide à la décision pour le diagnostic du Diabète**"

Ce chapitre se concentre sur l'application du Raisonnement à partir des Cas (RàPC) dans un système d'aide à la décision pour le diagnostic du diabète. Il explore les différents types de systèmes RàPC en médecine, leurs avantages et leurs limites. Des exemples concrets sont présentés pour illustrer l'utilisation de ces systèmes dans le domaine médical, y compris le diagnostic du diabète. Ce chapitre offre ainsi une compréhension approfondie de l'intégration du RàPC dans les systèmes d'aide à la décision pour le diagnostic de diabète.

Le quatrième chapitre “**Le framework jCOLIBRI**” Dans ce chapitre , nous allons passer en revue les assistants et les principaux outils proposés par Colibri Studio. Nous allons également explorer les fonctionnalités essentielles ainsi que l’architecture de jCOLIBRI, en mettant l’accent sur ses interfaces principales.

Le cinquième chapitre “**Conception**” Au sein de ce chapitre, nous nous concentrons sur l’étape cruciale de la conception et de l’étude de cas pour notre mémoire portant sur le développement d’un système d’aide à la décision en matière de diagnostic du diabète. Après avoir effectué une analyse détaillée de la méthodologie du Raisonnement à partir des Cas (RàPC) et étudié en profondeur le framework de développement jCOLIBRI, notre objectif principal est d’appliquer cette méthodologie afin de concevoir une solution pratique permettant le diagnostic du diabète.

Le sixième chapitre “**Implementation**” Dans cette section, nous effectuons une analyse détaillée de la mise en pratique de notre système. Notre objectif principal est de fournir une description approfondie des différentes étapes de développement, des choix de conception, des technologies utilisées, ainsi que des résultats obtenus. Nous présentons également notre application de manière détaillée.

Enfin, notre mémoire sera clôturé par une conclusion générale qui récapitulera l’ensemble de notre travail.

# Chapitre 1

## Les système d'aide à la décision médicale

### 1.1 Introduction

Chaque jour, nous sommes amenés à prendre des décisions qui peuvent être insignifiantes comme choisir notre petit-déjeuner ou importantes comme un diagnostic médical par exemple. Dans tous les cas, il est préférable de prendre des décisions qui mènent à de bons résultats plutôt que de mauvais résultats.

”La prise de bonnes décisions” signifie que nous sommes informés et avons des informations pertinentes et appropriées sur lesquelles baser nos choix parmi les alternatives. Dans certains cas, nous appuyons les décisions à l'aide de données existantes et historiques, tandis que d'autres fois, nous collectons l'information, en particulier pour un processus de choix particulier. Les informations se présentent sous forme de faits, de chiffres, d'impressions, de graphiques, d'images et de sons. Elles doivent être collectées à partir de différentes sources, assemblées et organisées.

### 1.2 La prise de décision

La théorie de la prise de décision de Simon est un cadre qui permet d'avoir une vision plus réaliste du monde. Les décisions prises ont un impact sur les prix et les sorties. Selon Simon (1977) [19] prendre une décision revient à choisir entre des cours d'action alternatifs. Cela peut même signifier choisir entre l'action et la non-action. Contrairement aux théoriciens classiques, Simon suggère qu'il n'y a jamais un seul meilleur cours d'action ou de décision [3]. C'est parce qu'on ne peut pas avoir des informations complètes sur quelque chose, il y aura donc toujours un meilleur cours d'action ou de décision.

#### 1.2.1 Le processus de la prise de décision

On peut considérer le processus de prise de décision comme un parcours qui commence par les données et avance vers les procédures de décision [1]. Nous trouvons, plusieurs processus tel que celui proposé par Mintezberg et al [24], ou celui proposé par Tsoukias [35]. Mais le processus le plus connu est celui de Simon.

Simon (1977) [31], a identifié la prise de décision comme un processus en trois phases majeures : l'intelligence, la conception et le choix, il a ensuite ajouté une quatrième phase, implémentation [3]. Le modèle de Simon est la caractérisation la plus concise et complète de la prise de décision rationnelle.

Une image conceptuelle du processus de prise de décision est présentée dans la figure 1.1. Il y a un flux continu d'activité de l'intelligence à la conception au choix, mais à n'importe quelle phase, il peut y avoir un retour à la phase précédente (rétroaction)[3] .

### 1.2.2 Les phases du processus de prise de décision

La première phase du processus décisionnel, appelée "l'intelligence", consiste à identifier un problème ou une opportunité. Le décideur rassemble des informations de l'environnement et évalue les performances de l'organisation en termes d'objectifs. Cela peut impliquer la comparaison avec d'autres organisations ou l'évaluation des activités internes. Cette phase est essentielle pour prendre des décisions éclairées.[3]

La deuxième phase est la conception. À cette phase, le décideur définit le choix particulier à faire. Il établit les objectifs spécifiques à considérer dans un contexte de choix particulier et identifie les alternatives appropriées. Cette étape inclut généralement la définition des alternatives, la collecte de données, la modélisation et l'examen des facteurs qui pourraient ne pas s'insérer dans le modèle.[3]

Dans la troisième phase, le décideur examine les informations, compare les alternatives, choisit les meilleures alternatives et évalue ce choix pour sa sensibilité aux hypothèses. Le but du SAD est de rassembler une intelligence d'affaires appropriée et des modèles pour aider l'individu à considérer un problème ou une opportunité sous différents angles avec de meilleures informations.[3]

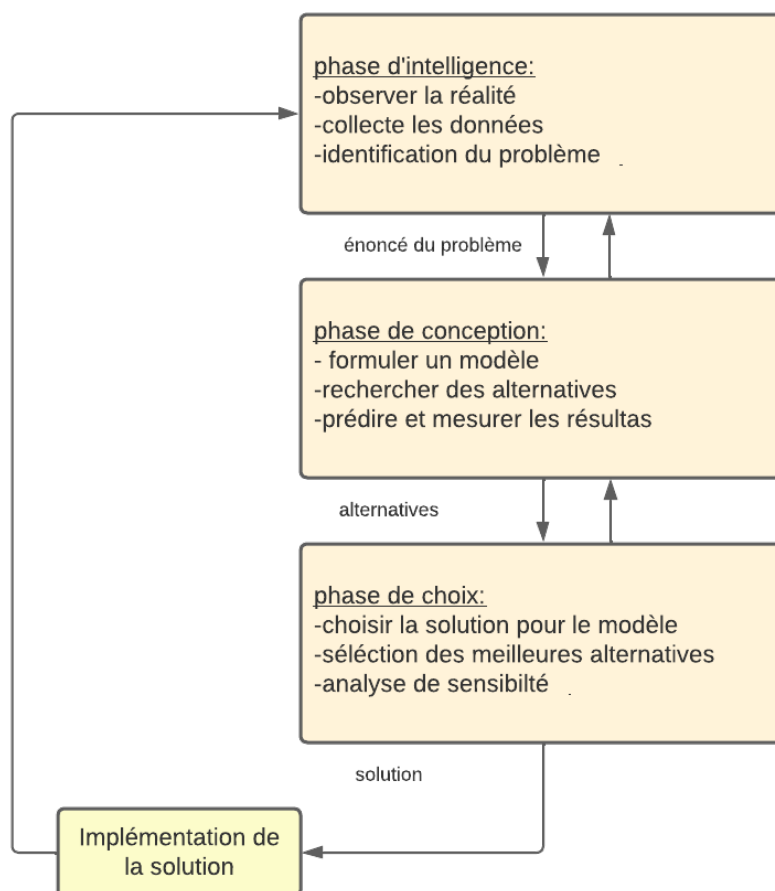


FIG. 1.1 : Les phases du processus de prise de décision[3].

### 1.3 Les systèmes d'aide à la décision

Un SAD est un système informatique qui soutient la prise de décision en aidant le décideur à organiser les informations et à modéliser les résultats. Et selon Little (1970) [3] « le SAD un ensemble de procédures basées sur des modèles pour traiter les données et les jugements afin d'aider un gestionnaire dans sa prise de décision ». Il soutient que pour réussir un tel système, il doit être simple, robuste, facile à contrôler, adaptatif, complet sur les questions importantes et facile à communiquer [3]. C'est pourquoi les SAD s'avèrent nécessaires pour fournir aux décideurs des informations pertinentes qui leur permettront de mieux comprendre les situations complexes.

#### 1.3.1 L'utilisation des SAD

Les systèmes d'aide à la décision peuvent être utilisés dans de nombreux domaines, tels que :

- **La gestion d'entreprise** : Les SAD aident les décideurs à analyser les données commerciales, à identifier les tendances, à évaluer les risques et à prendre des décisions éclairées pour améliorer l'efficacité et la rentabilité de l'entreprise.
- **Environnement** : Les SAD peuvent être utilisés dans la gestion de l'environnement pour la modélisation des écosystèmes, l'évaluation des impacts environnementaux, et la gestion des ressources naturelles.
- **Finance** : Les SAD peuvent aider à l'analyse de portefeuille, à la gestion des risques, à la prévision des marchés financiers, à l'évaluation des investissements et à la gestion des actifs
- **Santé** : Les SAD sont utilisés dans le domaine de la santé pour aider les médecins à prendre des décisions cliniques. Ils peuvent analyser des symptômes, des antécédents médicaux, des résultats d'examens et des données génétiques pour proposer des diagnostics, des traitements et des recommandations thérapeutiques.

#### 1.3.2 Classification des SAD :

Il y a plusieurs manières de classer les SAD, donc nous avons choisi de présenter deux classifications : celle de Hackathorn et Keen (1981) et celle de Efraim et al. (2001) ; Power (2007). [26]

Hackathorn et Keen (1981)[26] ont classé les SAD en trois groupes en fonction de leur capacité à fournir un soutien à toutes les étapes du processus de prise de décision, que ce soit pour un individu, un groupe ou une organisation entière

- Les SAD individuels qui fournissent de l'aide à une seule personne (décideur).

- Les SAD pour groupe qui fournissent un soutien à un groupe de personnes engagé dans des tâches distinctes mais étroitement liées.
- Les SAD pour organisation qui offrent un soutien à un groupe de personnes coopérant entre eux afin de réaliser des tâches ou des activités organisationnelles.

Efraim et al. (2001) et [Power (2007) [26] ont proposé une autre classification des SAD basée sur leur mode de fonctionnement, qui comprend des SAD axés sur les modèles, les données, les communications, les connaissances ou basés sur le Web :

- **les SAD guidés par des modèles :** Ce type de SAD utilise des données et des paramètres limités afin d'aider les décideurs à analyser une situation. Les SAD guidés par des données : Ce type de SAD capable d'analyser de vastes quantités de données diverses afin de générer des prédictions, des classifications, des associations et d'autres informations précieuses, etc.
- **Les SAD guidés par des communications :** Ce type de SAD utilise les technologies de réseau et de communication pour faciliter la collaboration et la communication pour la prise de décision.
- **Les SAD à base de connaissances :** Ce type de SAD repose sur divers types d'expertise pour aider à résoudre les problèmes. L'expertise implique une connaissance approfondie d'un domaine spécifique qui peut prendre différentes formes telles que des règles, des procédures, des recommandations, et d'autres connaissances similaires.
- **Les SAD guidés par le web :** Ce type de SAD implémenté dans un environnement distribué et il est spécialement adapté pour aider les clients en ligne à prendre des décisions éclairées sur l'achat de produits ou services. En fournissant des informations supplémentaires, il aide à faciliter le processus de prise de décision.

### 1.3.3 La combinaison de l'IA et des systèmes d'aide à la décision

Une nouvelle forme d'assistance informatique pour les décideurs est apparue grâce à l'intégration de l'IA et des systèmes d'aide à la décision. Cela a permis aux décideurs d'améliorer leurs capacités de raisonnement dans des environnements complexes.

De nombreux chercheurs ont étudié en profondeur les systèmes d'aide à la décision dotés "d'intelligence", ainsi que l'expertise du domaine. Les techniques intelligentes sont utiles pour analyser les données et fournir des prévisions, quantifier l'incertitude, fournir facilement des informations et suggérer la marche à suivre. [1]

La figure 1.2 souligne comment la prise de décision est influencée par une gamme variée de techniques intelligentes.

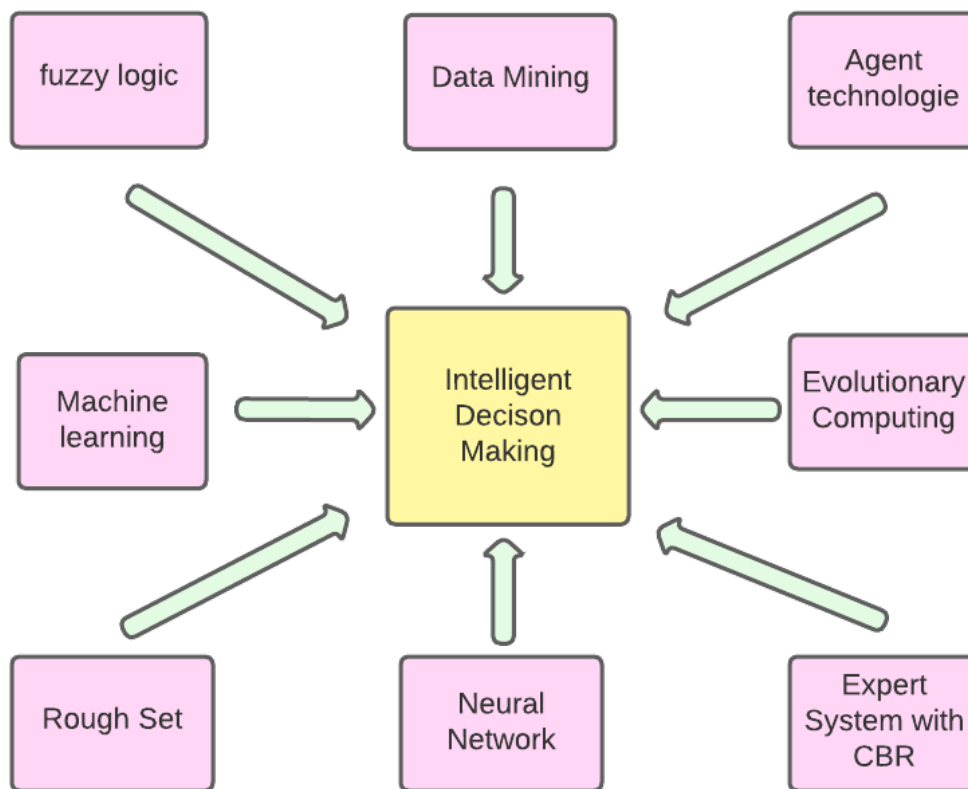


FIG. 1.2 : Techniques intelligentes en aide à la décision[1]

### 1.3.4 La décision médicale

Au centre de l'acte médical se trouve la décision médicale, qui implique de poser un diagnostic, de proposer ou de différer un traitement, entre autres. Pour aider le personnel de santé à prendre des décisions éclairées, de nombreuses applications d'aide à la décision ont été développées dans ce domaine. Cela implique l'utilisation de divers outils d'aide à la décision. Dans le cadre de ce domaine, l'aide à la décision consiste en l'extraction de connaissances dans de grands volumes de données pour en extraire des informations pertinentes [16].

Cependant, la prise de décision en médecine générale est un processus complexe qui implique la reconnaissance des différentes composantes : composantes biomédicales (objectives) ainsi que d'autres composantes, plus subjectives [4]. Pour faciliter ce processus, des systèmes ont été conçus et développés dans le but d'assister les professionnels de santé dans leur prise de décision. Ce qui est communément appelé : aide à la décision médicale.

### 1.4 Le système aide à la décision médicale

Aujourd'hui, les médecins ont accès à une grande quantité de données sur chaque patient, mais cette énorme quantité de données peut parfois causer des difficultés dans la prise de décision. Selon les exigences de la méthodologie de la médecine factuelle [13], un médecin devrait prendre des décisions cliniques en se basant sur les meilleures recherches disponibles. Cela est possible grâce à l'utilisation de systèmes informatiques appropriés.

#### 1.4.1 Définition

Il y a plusieurs définitions concernant les SADM ont été proposées par exemple : Selon Serroussi et Bouaud, [33] les systèmes d'aide à la décision médicale (SADM) sont des applications informatiques dont le but est de fournir aux cliniciens en temps et lieux utiles les informations décrivant la situation clinique d'un patient ainsi que les connaissances appropriées à cette situation, correctement filtrées et présentées afin d'améliorer la qualité des soins et la santé des patients.

Berner précise aussi que pour « améliorer la qualité des soins et la santé des patients les informations doivent également être : correctement filtrées et présentées ; fournies en temps et lieu utiles » [25] [1].

#### 1.4.2 La typologie des SADM

Les objectifs d'un système dépendent des utilisateurs, de leurs fonctions et de leurs connaissances. Il existe trois grandes catégories de systèmes d'aide :

- **Les systèmes d'assistance documentaire**

Les systèmes qui offrent une assistance documentaire sont des aides indirectes à la prise de décision. Ces systèmes visent à fournir un accès rapide aux informations pertinentes, mais n'ont pas de méthode de raisonnement à proprement dit mais ils doivent gérer des bases de données. Le système d'aide au diagnostic médical est actuellement utilisé toujours dans sa fonction documentaire. [4]

- **Les systèmes de rappels automatiques ou systèmes d'alerte**

Les médecins peuvent utiliser ces systèmes comme rappel des erreurs à éviter ou comme des facteurs importants à prendre en compte lors de la prise de décision. Les systèmes n'aident pas au raisonnement ou à la compréhension générale du cas du patient, mais plutôt une aide-mémoire utile fournissant une information utile et pertinente pour des situations cliniques plus ou moins simples. Par exemple, une mise en garde lors de la description d'un dosage de médicament peut être considérée comme une aide précieuse à la prise de décision. [1]

- **Les systèmes consultants**

L'objectif de ces systèmes est de fournir un avis d'expert sur une situation clinique précise, qu'elle soit de nature diagnostique ou thérapeutique. Les systèmes experts médicaux peuvent être classés dans cette catégorie, car ils raisonnent sur des scénarios médicaux prédéfinis et fournissent à l'utilisateur des conclusions étayées par des preuves basées sur les méthodes de raisonnement employées. C'est dans cette catégorie que les progrès les plus importants ont été réalisés en termes de systèmes d'aide à la décision [1].

### 1.4.3 Structure d'un SADM

Les systèmes D'aide à la décision médicale se composent de [25] [1] :

- une base de connaissances construite : c'est la partie du système qui stocke les connaissances médicales et les règles de raisonnement. Elle peut inclure des informations sur les maladies, les symptômes, les traitements et les médicaments, ainsi que des règles de décision pour aider à déterminer un diagnostic ou un traitement.
- L'inférence utilisateur : cette partie du système permet à l'utilisateur de communiquer avec la base de connaissances et d'entrer des informations sur un patient, comme les symptômes, les antécédents médicaux et les résultats de tests. L'interface peut être une application logicielle, un site web ou une interface vocale.
- Le moteur de raisonnement : c'est la partie du système qui utilise les règles de décision stockées dans la base de connaissances pour raisonner sur les informations fournies par l'utilisateur. Le moteur de raisonnement peut générer des recommandations pour un diagnostic ou un traitement, ou présenter des options à l'utilisateur pour l'aider à prendre une décision éclairée.

### 1.4.4 Les principales fonctions du SADM

Alertes et rappels sont les fonctions d'aide à la décision les plus courantes dans les systèmes d'aide à la décision médicale. Dans un environnement en temps réel, ces fonctions d'aide à la décision sont intégrées aux dispositifs de surveillance pour fournir des notifications immédiates lorsque certaines conditions sont remplies. Par exemple, dans un environnement de soins aigus, les moniteurs d'oxygène et de pression artérielle peuvent alerter les infirmières lorsque l'état d'un patient dépasse une valeur seuil prédéfinie. Dans un environnement de soins chroniques, les décideurs peuvent recevoir des notifications par e-mail ou pager des résultats de laboratoire anormaux. Certains systèmes d'aide à la décision médicale fournissent même des fonctions de reconnaissance et d'interprétation d'images, qui sont particulièrement utiles dans les grands hôpitaux où les rapports de radiologie peuvent être interprétés et les experts peuvent être alertés.[1]. [22]

Fonction	Exemple de problèmes cliniques
<b>Alerte</b>	Basée sur les résultats de laboratoire avec différents niveaux personnalisables
<b>Diagnostic</b>	Identifier le diagnostic possible en fonction de l'historique, du physique, des résultats et données saisies
<b>Rappel</b>	Rappeler aux praticiens les ordres et leurs calendriers
<b>Notification</b>	Non-conformité, risques, événements anormaux et périodes de soins
<b>Suggestion</b>	Ajustements médicamenteux, les tendances et les dosages actuels de médicaments
<b>Interpretation</b>	Directives pour la situation actuelle - calendrier de tests-laboratoire, protocoles de soins
<b>Prédiction</b>	Predire les résultats en fonction de certaines variables indépendantes
<b>Assistance</b>	Fournir un autre médicament suite à une interaction médicamenteuse ou à une allergie
<b>Critique</b>	L'utilisation d'une procédure médicale sur la base des directives médicales applicables en médecine et des antécédents médicaux du patient

TAB. 1.1 : Fonction d'aide à la décision et exemple de problèmes cliniques[22]

### 1.4.5 Les objectifs du SADM

Les objectifs des systèmes d'aide à la décision médicale peuvent être résumés en quelques points[1] :

- Fournir un accès rapide et efficace à l'information médicale pertinente.
- Améliorer la qualité des décisions prises par les professionnels de santé.
- Réduire les erreurs médicales et les risques pour les patients.
- Améliorer l'efficacité et l'efficience du processus de prise de décision médicale.
- Faciliter la communication et la collaboration entre les différents professionnels de santé impliqués dans la prise en charge d'un patient.
- Améliorer la satisfaction des patients en leur offrant des soins plus personnalisés et de meilleure qualité.

### 1.5 Conclusion

En conclusion, les systèmes d'aide à la décision médicale sont devenus une partie intégrante de la prestation de soins de santé aujourd'hui. Ces systèmes ont diverses fonctions qui aident les prestataires de soins de santé à prendre de meilleures décisions, à fournir des soins plus efficaces et à réduire les erreurs médicales. De la simple documentation et stockage d'informations aux algorithmes sophistiqués et aux fonctions de reconnaissance d'images, les systèmes d'aide à la décision médicale ont évolué considérablement au fil des ans. Malgré leurs nombreux avantages, ces systèmes sont encore confrontés à des défis tels que la confidentialité et la sécurité des données, la nécessité d'une formation continue des utilisateurs et le coût élevé de la mise en œuvre. Néanmoins, avec le développement et l'amélioration continus, les systèmes d'aide à la décision médicale ont le potentiel de révolutionner les soins de santé et d'améliorer les résultats pour les patients.

## Chapitre 2

### Raisonnement à partir de cas

### 2.1 Introduction

L'homme possède une remarquable aptitude à tirer des leçons de ses expériences et à s'adapter aux nouvelles situations. Face aux défis quotidiens, nous nous tournons instinctivement vers nos expériences passées pour trouver des solutions efficaces. Nous utilisons nos souvenirs de situations similaires vécues auparavant, que nous comparons ensuite à la situation présente afin de générer de nouvelles solutions.

Cette capacité de raisonnement basée sur des cas est cruciale pour notre survie en tant qu'espèce. Au fil des siècles, nous avons évolué en développant des compétences de plus en plus sophistiquées pour nous adapter aux changements de notre environnement. En utilisant le raisonnement basé sur des cas, nous enrichissons notre expérience et renforçons notre capacité à résoudre avec succès les problèmes futurs.

Raisonnement à Partir de Cas, est également une méthode d'Intelligence Artificielle qui consiste à résoudre des problèmes en utilisant des connaissances stockées dans des bases de données de cas antérieurs. Cette méthode est largement utilisée dans divers domaines tels que la médecine, l'ingénierie, la finance et la recherche.

### 2.2 Bref historique

Le Raisonnement à partir des Cas (CBR) est un concept qui remonte à 1977 avec les travaux de Schank et Abelson. Il repose sur l'idée que notre connaissance des situations est stockée sous forme de scripts, permettant de créer des attentes et de tirer des inférences. Janet Kolodner a développé le premier système CBR, CYRUS, basé sur le modèle de mémoire dynamique de Schank. Bruce Porter a proposé une approche alternative avec le système PROTOS, unifiant la connaissance générale du domaine avec la connaissance spécifique des cas. Dans le domaine juridique, le système HYPO utilise des cas juridiques pour interpréter des situations et produire des arguments. En Europe, des recherches ont été menées par Sleeman, Keane, Richter, Althoff, et Aamodt, qui ont appliqué le CBR dans des domaines tels que le diagnostic complexe, le génie civil et l'apprentissage. Le CBR est également appliqué en Inde et au Japon. L'intérêt pour le CBR augmente dans le monde, avec des applications commerciales et une publication croissante d'articles spécialisés en IA.[38]

### 2.3 Raisonnement à partir de cas

Le raisonnement basé sur des cas connu sous l'acronyme RàPC, ( en Anglais CBR : Case-Based Reasoning) est une technique qui vise à émuler le raisonnement humain de type analogique en l'implémentant dans des systèmes informatiques. [18] Le RàPC est une approche de résolution de problèmes en intelligence artificielle qui s'appuie sur des exemples concrets. Au lieu de recourir à des règles ou à des modèles préétablis, le RaPC utilise des cas antérieurs similaires pour résoudre de nouveaux problèmes. Les cas sont des représentations de situations passées qui contiennent à la fois la description du problème et sa solution. En utilisant ces cas comme base de connaissances, le RaPC peut extraire des informations pertinentes, effectuer des comparaisons et adapter les solutions aux nouveaux

problèmes rencontrés. Cette approche permet d'exploiter efficacement l'expérience passée pour guider la prise de décision et améliorer la résolution de problèmes.[20] Chaque cas est représenté par une paire "problème" (ou cas cible) et "solution", stockée dans une base de cas. Chaque cas décrit une situation particulière et est indépendant des autres cas.[32]

### 2.4 Connaissance de système à partir de cas

Le processus de raisonnement à partir de cas est constitué de quatre (4) modules de connaissances à savoir la base de cas ,le vocabulaire d'indexation, la mesure de similarité et les connaissances d'adaptation :[23]

- **La base de cas** : C'est une collection de cas antérieurs stockés sous forme de descriptions textuelles ou de données structurées. Cette base de cas est utilisée pour rechercher des cas similaires qui peuvent aider à résoudre un nouveau problème.
- **Le vocabulaire d'indexation** :Ce module de connaissances contient les informations requises pour décrire un cas en utilisant un ensemble fini de caractéristiques, ce qui est essentiel pour la phase de récupération des cas et pour la construction de la base de cas. En effet, il a une incidence significative sur ces deux étapes du processus.
- **La mesure de similarité** : C'est une fonction qui mesure la similitude entre deux cas. Cette mesure est basée sur les caractéristiques ou les attributs des cas qui ont été identifiés dans le vocabulaire d'indexation. Plusieurs mesures de similarité peuvent être utilisées, comme la distance euclidienne, la corrélation de Pearson, ou encore la similarité cosinus.
- **Les connaissances d'adaptation** : Ce module permet d'adapter les solutions trouvées dans la base de cas aux nouvelles situations. Il s'agit d'un processus de modification des solutions existantes pour les rendre plus adaptées à la situation actuelle. Cette étape peut nécessiter une certaine expertise ou des connaissances supplémentaires pour adapter la solution au contexte spécifique.

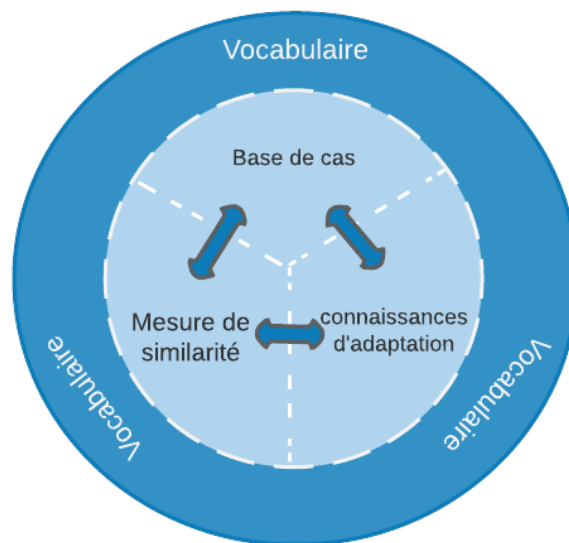


FIG. 2.1 : Conteneur de connaissance.  
[23]

La figure : 1 expose les quatre modules de connaissances en illustrant leurs liens mutuels. Le module vocabulaire occupe une position centrale en tant que pilier des autres modules, et les flèches matérialisent les interdépendances entre ces derniers.

## 2.5 Description générale du cas

### 2.5.1 Définition du cas

Un cas est une expérience qui est conceptualisée en tant que connaissance et qui peut servir de guide pour le système de RàPC lors de la résolution de problèmes variés. La nature des informations contenues dans un cas dépend de son domaine d'application et de ses objectifs spécifiques.[27] Généralement un cas comprend :[38]

- **Un problème** : qui décrit la situation initiale, le contexte et l'état du monde dans lequel le cas s'est produit. Cela peut inclure des informations sur les symptômes, les conditions environnementales, les besoins ou les objectifs des utilisateurs, etc..
- **Une solution** : elle décrit la manière dont le problème a été résolu dans le cas, en décrivant les étapes, les actions et les décisions qui ont été prises pour atteindre l'objectif ou résoudre le problème.
- **Un résultat** : il décrit l'état final du monde après l'application de la solution.

### 2.5.2 Représentation des cas

Pour utiliser le raisonnement à partir de cas (CBR), il est essentiel de savoir comment représenter un cas. La représentation d'un cas consiste à déterminer les attributs qui le décrivent, la structure qui permet de décrire son contenu, ainsi que l'organisation des cas dans la base de cas.[30]

Il est possible qu'un cas puisse être présenté sous l'une des formes suivantes :[30]

- **La représentation vecteur** : la forme la plus simple de représentation de cas, dans laquelle chaque cas est représenté par un ensemble de caractéristiques décrivant le problème et la solution associée. Tous les cas ont le même type et le même nombre de caractéristiques, et la similarité est directe, sans relation ou contrainte entre les caractéristiques.
- **La représentation frame ou entité** : offre une structure claire et concise pour représenter une connaissance en une seule entité. Chaque entité est constituée d'un nom et d'un ensemble d'attributs. Ces attributs peuvent être une valeur simple ou un pointeur vers une autre entité. Cette méthode permet de combiner toutes les connaissances nécessaires d'un concept en une seule entité, ce qui facilite la compréhension et la gestion des connaissances.
- **La représentation textuelle** : qui est utilisée lorsque toutes les connaissances sont disponibles sous forme de texte. Cette méthode vise à utiliser les données textuelles de manière automatique ou semi-automatique pour améliorer le processus de résolution des problèmes en comparant des cas similaires.
- **La représentation prédicat** : utilise des relations entre les attributs d'un cas, représentées par une partie condition "Si" et une partie action "Alors" ou par plusieurs prédicats, permet d'utiliser à la fois des règles et des faits pour obtenir une représentation plus complète et précise de la connaissance.

### 2.5.3 Indexation des cas

Pour mieux organiser les cas dans la base de cas et faciliter leur recherche ultérieure, il est nécessaire de les indexer.

L'indexation de cas consiste à leur attribuer des indices afin de faciliter leur recherche ultérieure. Les chercheurs en RàPC ont recommandé des directives pour l'indexation de cas, qui impliquent de choisir des indices "prédicatifs" répondant aux objectifs du cas et des indices "abstraites" pour une utilisation future étendue et pour reconnaître les cas ultérieurement.[38]

Les méthodes d'indexation peuvent être manuelles ou automatisées. Dans le cas de l'indexation manuelle, il est important de bien définir les objectifs d'utilisation des cas ainsi que les circonstances dans lesquelles ils seront utilisés. Cependant, l'indexation automatisée est de plus en plus répandue et il existe actuellement trois méthodes couramment utilisées[38] :

- **Les méthodes d'indexation basées sur la similarité** : générer un ensemble d'indices pour un cas abstrait en se basant sur les caractéristiques communes des cas similaires.
- **L'indexation basée sur l'explication** : analyse les cas pour trouver les attributs prédictifs qui peuvent être utilisés comme indices pertinents pour faciliter la récupération des cas ultérieurement.
- **Les méthodes basées sur l'apprentissage inductive** : utilisent des algorithmes tels que ID3 et C4.5 pour choisir les attributs qui peuvent servir d'indices.

### 2.5.4 Stockage des cas

Lorsqu'on conçoit des systèmes CBR efficaces, il est crucial de considérer la façon dont les cas sont stockés, en veillant à ce que cela reflète la vue conceptuelle de ce qui est représenté dans le cas et en prenant en compte les indices qui le caractérisent.

La base de cas peut être stockée dans divers formats tels que des bases de données, fichiers XML ou fichiers texte. Les cas sont indexés pour une recherche efficace. Elle doit être organisée de manière à faciliter des méthodes de recherche et de récupération efficaces.

La mémoire de tous les cas précédemment enregistrés constitue la base de cas du système de raisonnement à partir de cas. Lors de la création d'une telle base, trois domaines généraux doivent être pris en compte :[38]

- La structure et la représentation des cas.
- Le modèle de mémoire utilisé pour organiser l'ensemble de la base.
- La sélection des indices permettant l'identification de chaque cas.

Il est important que les modèles de stockage de cas trouvent un équilibre entre la préservation de la richesse sémantique des cas et de leurs indices, d'une part, et la simplicité de l'accès et de la récupération des cas pertinents, d'autre part.

Il existe deux modèles de mémoire et d'organiser des cas les plus influents qui sont :le modèle de mémoire dynamique de Schank et Kolodner, ainsi que le modèle catégorie-exemple de Porter et Bareiss.[38]

## 2.6 Le cycle de RàPC

Le RàPC est généralement décrit par un cycle de quatre phases de haut niveau pour la résolution des problèmes :Récupération des cas similaires(RETRIEVE), l'adaptation ou la réutilisation des cas trouvés(REUSE), la Révision des solutions(REVISE) et la Rétention (RETAIN)[27]

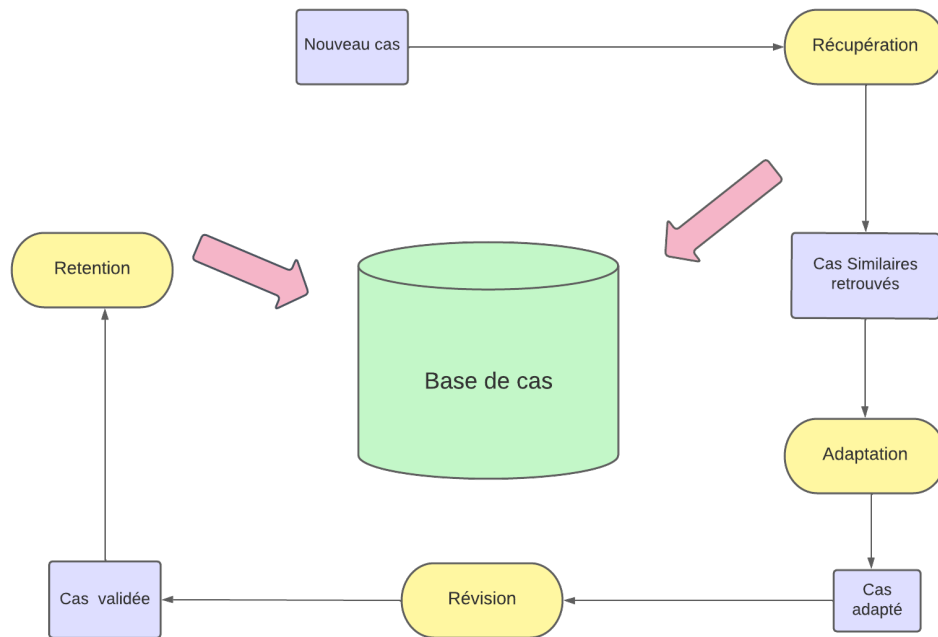


FIG. 2.2 : Le cycle du RàPC  
[27]

### 2.6.1 Phase de la Récupération des cas similaires (RETRIEVE) :

La phase de la recherche des cas similaires (RETRIVE) se déroule en deux tâches principaux qui sont : le filtrage et la sélection[15]

La mise en œuvre de ces tâches est conditionnée par la représentation et l'organisation des cas, ainsi que par leur indexation dans la base de cas.

#### La tâche de filtrage :

le filtrage a pour objectif de limiter le nombre de cas à considérer lors de la sélection. Bien que cette étape puisse être omise, elle est souvent nécessaire pour des raisons de performance. Les algorithmes de filtrage peuvent varier selon le type de représentation des cas utilisé, et il existe plusieurs approches possibles.[15]

#### La tâche de sélection

La sélection des cas appropriés est effectuée en examinant la base de cas pour trouver les cas les plus similaires au cas cible. Cette étape consiste à rechercher dans la base de cas les cas sources qui sont les plus proches du cas cible, afin de sélectionner les meilleurs cas possibles.[27]

Pour effectuer cette recherche, des algorithmes spécifiques sont utilisés pour trouver les cas similaires à celui qui est étudié. Des heuristiques sont également employées pour évaluer la similarité entre le cas cible et les cas sources disponibles dans la base de cas. Cette tâche peut être réalisée indépendamment de la tâche de filtrage préalable, si celle-ci a été omise.[15]

### Algorithme de recherche des cas similaires

Pour la recherche des cas adéquats, il existe certains nombres algorithmes, mentionnons quelques exemples :

**Les K plus proches voisins (KPPV) :** est une méthode pour estimer la similarité entre le cas cible et le cas source, dont le K est le nombre de cas sources voisins qui sont considérés comme étant proches du cas cible.

**Les approches inductives :** l'approche inductive implique l'analyse des données pour identifier et extraire les facteurs les plus importantes et les plus efficaces pour discriminer les cas [38], et utilise ces facteurs des problèmes similaires pour générer de nouvelles connaissances.

L'approche inductive souvent utilisée dans le cadre de l'arborescence pour la prise et l'amélioration de décision à l'aide d'algorithmes d'arbre de décision tels que ID3 et CART. Effectivement, cette approche engendre une structure de type arbre de décision pour organiser les cas en mémoire.[27]

**L'induction guidée par la connaissance :** Cette méthode applique des connaissances au processus d'induction en déterminant manuellement les caractéristiques du cas qui sont connues ou censées affecter la caractéristique initiale du cas.[38]

**La recherche basée sur la structure :** la recherche renvoie tous les cas qui répondent à certains paramètres[27]. Cette technique est fréquemment appliquée en préalable à les autres techniques, afin de réduire le champ de recherche à une section adéquate de la base de données de cas.[38]

### Mesures de similarité

Il est possible d'identifier plusieurs mesures qui sont liées à la représentation des cas. On peut les regrouper en deux catégories principales[27] :

- **Les mesures de similitude** qui indiquent le niveau de similitude entre deux cas.
- **Les mesures de dissimilitude** qui indiquent le niveau de différence entre deux cas.

La similarité peut être mesurée à différentes échelles : localement ou globalement.[27]

**La similarité locale :** Les mesures de similarité locale sont basées sur les propriétés des cas, qui peuvent être de différents types tels que numériques, symboliques ou taxonomiques. Le type de propriété joue généralement un rôle majeur dans la mesure de la similarité [27]

- pour les valeurs de numériques :

$$\text{sim}(a, b) = 1 - \frac{|a - b|}{\text{range}}$$

- pour les valeurs symboliques (mono-valeurs) :[5]

$$\text{sim}(a, b) = \begin{cases} 1 & \text{pour } a = b \\ 0 & \text{pour } a \neq b \end{cases}$$

- pour les valeurs symboliques (multi-valeurs) :[5]

$$\text{sim}(a, b) = \frac{\text{card}(a) \cap \text{card}(b)}{\text{card}(a \cup b)}$$

- pour les valeurs taxonomiques :

$$\text{sim}(a, b) = \frac{h(\text{commonnode}(a, b))}{\text{Min}(h(a), h(b))}$$

Où :

$a$  et  $b$  : sont des valeurs des descripteurs.

card : est la cardinalité de l'ensemble.

range : est la valeur absolue de la différence entre la borne supérieure et la borne inférieure de l'ensemble des valeurs.  $h$  : est le poids (le nombre de niveaux) de l'arbre taxonomique

### La similarité globale :

La similarité globale entre un cas cible et un cas source est calculée en utilisant les mesures de similarité locale. Il existe différents types de mesures de similarité globale qui sont choisis en fonction du domaine d'étude. Parmi ces types de mesures de similarité, on peut citer :[27]

-Weighted Block-City(Average)

$$\text{sim}(A, B) = \sum_{i=1}^n w_i \text{sim}_i(a_i, b_i)$$

- la distance de Minkowski : Si  $r = 2$  alors on retrouve la distance Euclidienne et si  $r = 1$  alors il s'agira de la distance de Manhattan.

$$\text{sim}(A, B) = \left[ \frac{1}{2} \sum_{i=1}^n \text{sim}_i(a_i, b_i)^r \right]^{\frac{1}{r}}$$

- Maximum based

$$\text{sim}(A, B) = \max_i w_i \text{sim}_i(a_i, b_i)$$

Où :  $n$  est le nombre d'attributs.

$w_i$  est le poids (évalué en fonction de l'importance) de l'attribut  $i$ .

$\text{sim}_i$  est la similarité locale calculée pour l'attribut  $i$ .

### 2.6.2 Phase de l'adaptation des cas trouvés(REUSE) :

Après avoir sélectionné les cas les plus pertinents lors de la phase de recherche, Le RàPC doit s'appuyer sur les solutions trouvées dans les cas sources identifiés pour pouvoir adapter la solution au nouveau problème. Il y a habituellement deux types d'adaptations[9] citewatson1994case :

**L'approche transformationnelle (ou structurelle) :** Nous exécutons promptement les règles

d'adaptation, en altérant astucieusement les solutions antérieures et en les réorientant de manière à résoudre le nouveau problème.

**L'approche générative (ou dérivationnelle) :** Nous préservons et appliquons les méthodes ou règles ainsi que les étapes qui ont engendré les solutions des cas précédents, afin de résoudre un nouveau problème .

### 2.6.3 Phase de la Révision des solutions(REVISE ) :

Lors de cette phase, le RàPC examine minutieusement la solution suggérée pour le cas analysé et évalue si elle répond de manière adéquate à la situation du cas. Si la solution ne répond pas aux exigences du cas, elle peut être ajustée .

Après la révision, la solution est évaluée à nouveau et jugée satisfaisante. Elle est ensuite considérée comme prête pour la phase suivante, où elle pourra être utilisée pour résoudre des problèmes semblables à l'avenir.[27]

La solution évaluée implique certaines actions, à savoir[27] :

- mettre en œuvre un test de solution suggérée dans le monde réel.
- effectuer une analyse exhaustive de la base de données de cas en utilisant toutes les descriptions de problèmes et de solutions afin de déterminer si des cas similaires ont été résolus de manière satisfaisante,.
- utiliser une autre méthode pour l'évaluation de la solution (simulateur, système expert classique etc..).

### 2.6.4 Phase de rétention (RETAIN) :

Lorsque les solutions aux problèmes sont évaluées et validées, la phase d'apprentissage devient cruciale pour le processus de raisonnement à partir de cas. Cette phase consiste à stocker les cas dans une base de données afin de les récupérer ultérieurement pour résoudre de nouveaux problèmes similaires. Cette étape permet d'enrichir la base de cas, ce qui augmente l'expérience du système et améliore sa capacité à résoudre des problèmes futurs. Par conséquent, la phase d'apprentissage est une étape fondamentale pour le processus de raisonnement à partir de cas.[27]

## 2.7 Quelques projets basés sur le RàPC

Ces projets montrent la polyvalence du RàPC et son utilisation dans des domaines variés, de la santé à l'industrie , ect.. [38]

**PROTOS** : [Porter Bareiss, 86] a été conçu pour la classification des troubles de l'ouïe en se basant sur les descriptions des symptômes, des antécédents médicaux et des résultats de tests des patients en audiologie clinique. Le développement de PROTOS a pris en compte les contraintes du monde réel, telles que la représentation limitée des connaissances et la disponibilité restreinte des données d'entraînement. Le système a été entraîné avec 200 cas regroupés en 24 catégories, et a atteint une précision de 100/100 après l'entraînement.

**PAKAR** : développé par Watson et Abdullah en 1994, est un système qui utilise le logiciel CBR-Express sous MS Windows pour identifier les causes potentielles de défauts

de construction et proposer des actions correctives appropriées. En intégrant des informations textuelles et des dessins de conception assistée par ordinateur, PAKAR est capable de proposer des conseils sur les causes probables des défauts et les solutions recommandées.

**CYCLOPS** : est un résolveur de problèmes de conception développé par Navinchandra en 1989, qui se focalise sur le domaine de l'aménagement paysager. Il vise à résoudre les problèmes de conception en identifiant les éventuels obstacles lors de la création de nouveaux équipements.

**CHEF** : élaboré par Hammond en 1986, vise à créer des recettes innovantes à partir de recettes déjà existantes. Pour y parvenir, CHEF commence par chercher une recette qui répond le plus possible à ses objectifs en cours. Ensuite, il utilise un ensemble de critiques d'objets et de règles de modification pour adapter l'ancienne recette et atteindre les buts de la nouvelle.

L'un des aspects clés de CHEF est sa capacité à expliquer les échecs par une description causale de leur occurrence. Il associe les recettes à l'explication des échecs afin de prédire les échecs dans des situations similaires. Enfin, CHEF enregistre les nouvelles recettes en les indexant en fonction des objectifs qu'elles remplissent et des problèmes qu'elles évitent.

**COACH** : élaboré par Collins en 1987, est destiné à travailler dans le domaine du football. Son but est de générer de nouveaux schémas de jeu en améliorant les plans stockés grâce à des opérations de débogage et de réparation. Bien que la bibliothèque de COACH contienne un nombre limité de schémas de jeu, le programme dispose de plusieurs stratégies d'adaptation pour créer de nouveaux schémas.

## 2.8 Domaines d'application

Les systèmes de RàPC, offrent une grande polyvalence d'application dans de nombreux domaines et pour différentes tâches à accomplir.

La médecine, le commerce, les services de consultation, la maintenance, le contrôle et l'analyse financière sont des secteurs où ces systèmes sont particulièrement développés.

Divers types d'applications sont envisageables, tels que la classification, le diagnostic, la planification, le design, l'interprétation, la justification, l'aide à la décision, la recherche de documents, la configuration ou encore la gestion des connaissances.

En fonction des domaines ou des types d'applications considérés, les modèles et méthodes des cas de raisonnement systémique peuvent varier. De fait, plusieurs auteurs ont proposé des typologies d'applications distinctes, en fonction de perspectives variées.[27]

## 2.9 Les avantages du raisonnement à partir de cas

La méthode de raisonnement à partir de cas (RàPC) offre de nombreux avantages par rapport à d'autres méthodes. Elle est plus facile à mettre en place que les approches basées sur des modèles de domaine tels que les bases de règles et permet de résoudre des

problèmes complexes pour lesquels il n'existe pas de solutions formelles ou logiques. Cette méthode est particulièrement utile pour les problèmes peu structurés ou mal définis, tout en évitant les problèmes d'acquisition de connaissances qui compliquent la conception de bases de connaissances de grande taille.

Grâce à sa capacité d'apprentissage à partir d'exemples et d'adaptation à de nouvelles situations, elle peut évoluer avec le temps et s'adapter à des environnements changeants. En stockant des cas passés et leurs solutions, cette méthode peut capitaliser sur l'expérience et réutiliser des solutions pour résoudre des problèmes similaires, réduisant ainsi les coûts et le temps de développement. De plus, elle peut traiter des problèmes en évolution rapide en utilisant de nouveaux cas pour adapter ses solutions. Enfin, elle est capable d'organiser et d'indexer efficacement de grandes quantités de données pour une recherche et une récupération rapides et efficaces.[11]

### 2.10 Conclusion

L'objet de ce chapitre était de clarifier les principes fondamentaux de la RàPC dans le contexte de l'IA. L'historique a permis de suivre l'évolution du RàPC en tant que paradigme pour la résolution de problèmes. D'innombrables recherches ont permis aux concepts du RàPC de mûrir, aboutissant à la mise en place de systèmes fondés sur cette méthode. Certains de ces systèmes ont été employés dans des projets de grande envergure. L'application de la RàPC s'étend à divers domaines, cependant, pour les besoins de notre projet, nous nous penchons sur l'exploitation de cette méthode pour les systèmes d'aide à la décision médicale. En effet, la complexité de la résolution de problèmes en médecine, notamment en termes de diagnostic, présente une similitude marquée avec le principe de la RàPC. Cette convergence a suscité plusieurs initiatives s'attaquant à une pluralité d'applications médicales grâce à la RàPC.

## Chapitre 3

# L'Application du RàPC dans un système d'aide à la décision pour le diagnostic du Diabète

## 3.1 Introduction

Le raisonnement du médecin pour résoudre les situations médicales dépend principalement de la probabilité que la situation actuelle ait été traitée auparavant, ce qui conduit le médecin à proposer une solution similaire à celle utilisée précédemment. Cette méthodologie de raisonnement est connue sous le nom de Raisonnement à partir de Cas (RàPC). En conséquence, énormément des travaux de recherche ont été menés dans le domaine médical sur ce mode de raisonnement, conduisant à la mise au point d'outils informatiques qui reposent entièrement sur le RàPC pour résoudre les problèmes de prise de décision. Ces recherches ont eu des implications pour divers domaines de l'intelligence artificielle, tels que la représentation des connaissances, la classification et les mesures de similarité, faisant du RàPC un mode de raisonnement complexe mais largement utilisé en prise de décision médicale.

## 3.2 Les types des systèmes RàPC en médecine

Il existe plusieurs types de systèmes de raisonnement à base de cas en médecine qui peuvent être orientés selon deux caractéristiques [1] :

### 3.2.1 Les caractéristiques orientées vers les objectifs

Ces caractéristiques peuvent aider les systèmes RàPC à fournir des recommandations plus personnalisées, plus précises et plus efficaces pour la prise de décision clinique en médecine par exemple :

- Les systèmes de diagnostic : Ces systèmes aident les médecins à diagnostiquer les maladies en fournissant des suggestions de diagnostic basées sur les symptômes du patient.
- Les systèmes de classification : ces systèmes tentent d'identifier le groupe auquel appartient un cas. Le système de classification d'images en est un exemple.
- Les systèmes de tutorat : les systèmes RàPC utilisent des exemples généralement des cas réels de médecine, pour l'apprentissage et permettent au système de fournir un tutorat. Par conséquent, un système de tutorat médical permet aux cliniciens l'accéder à des cas généralement réels, voire parfois fictifs, afin de faciliter l'apprentissage à travers des exemples pratiques.
- Les systèmes de planification : ces systèmes apportent une assistance dans la conception d'un plan ou d'un schéma comprenant thérapeutique plusieurs étapes.

### 3.2.2 Les caractéristiques orientées vers sa construction

Ces caractéristiques peuvent aider à garantir la qualité et la fiabilité des systèmes RàPC en médecine, en s'assurant qu'ils sont basés sur des données et des connaissances médicales précises et en garantissant qu'ils sont efficaces dans la pratique clinique.

- Les systèmes hybrides : l'objectif de ces systèmes hybrides est de combiner le raisonnement à partir de cas avec d'autres méthodologies de raisonnement pour obtenir une approche plus intégrée et efficace
- Les systèmes autonomes : Le degré d'autonomie est important pour un système de diagnostic. Il se fait donc en fonction du besoin de l'intervention du décideur dans le cycle de raisonnement et lors de l'évaluation des résultats.

### 3.3 Les limites de l'aide à la décision médicale par RàPC

Malgré utilisation du RàPC dans de nombreuses situations médicales appropriées, l'approche RàPC présente encore certaines limitations principalement liées au domaine médical, qui est un peu particulier à cause des types de données et des connaissances manipulées.

Gierl et al., citent quelques-unes de ces limites en médecine [17] [40] :

- Parfois, l'approche RàPC ne peut pas fournir une solution définitive à un problème, alors il proposera seulement un ensemble de solutions possibles
- Pour un système RàPC médical, une interaction utilisateur significative est nécessaire, notamment lors de l'évaluation des résultats.
- Comment formaliser et prendre en compte l'approche diagnostique ou thérapeutique des médecins
- Les systèmes de raisonnement à partir de cas nécessitent des bases de cas cliniques réelles pour répondre aux besoins spécifiques des cliniciens, car ils dépendent de ces références (cas exemples) pour leur fonctionnement.
- L'adaptation et l'utilisation de RàPC dans le domaine médical est une tâche hautement complexe et laborieuse en raison du grand nombre de descripteurs ou attributs qui caractérisent les cas médicaux.

### 3.4 Exemples de SADM fondés sur un RàPC

Voici quelques exemples de systèmes d'aide à la décision médicale fondés sur le raisonnement à partir de cas [6] :

**M-Ultranet**[14] : (Z. El Balaa, 2003; Z. El Balaa and Uziel, 2003) est un système d'aide au diagnostic à base de cas utilisé en gynécologie pour détecter précocement les malformations fœtales. Son objectif est d'aider les échographistes dans leur diagnostic en utilisant leur expérience avec des cas cliniques similaires déjà diagnostiqués. Les cas sont représentés par des objets et sont organisés hiérarchiquement en différents concepts et sous-concepts. Chaque concept est représenté par un ou plusieurs attributs, qui consistent en des connaissances liées à la structure anatomique fœtale, ainsi que des données cliniques sur le patient et des connaissances appropriées du domaine. Pour mesurer la similitude entre les cas, des mesures médicales spécifiques développées par des gynécologues sont utilisées.

**SFDA** [7] : (Chakraborty et al., 2011) est un système d'aide au diagnostic à base de cas utilisé pour le diagnostic de la grippe porcine. Le système prédit si le patient est atteint de la grippe porcine ou non en fonction des symptômes présentés par le patient et de son âge. Chaque cas dans la base de cas est représenté par un ensemble d'attributs (âge, température, nausée, etc.) et chaque attribut est associé à un coefficient de pondération. Les cas les plus similaires sont récupérés à l'aide d'une fonction de similarité qui détermine le degré de similitude entre le cas cible et chaque cas de la base de données en se basant sur l'algorithme K-NN (Nearest Neighbor Algorithm).

**MedCase** [6] : (Wang and Tansel, 2013) est une plateforme open source qui permet aux professionnels de la santé de concevoir des systèmes SADM distribués basés sur RàPC. Les cas sont représentés à l'aide d'ontologies, ce qui facilite la construction et le partage de cas pour les systèmes distribués.

Le développement de MedCase s'appuie sur des recherches antérieures présentées dans (Hsien-Tseng Wang, 2013) où les auteurs proposent la construction d'un système d'aide au diagnostic médical basé sur RàPC en utilisant la technologie du web sémantique comme représentation de la connaissance.

## 3.5 Le RàPC comme un système d'aide à la décision pour le diagnostic du Diabète

L'objectif de ce travail de recherche était de développer un système basé sur le raisonnement à base de cas pour le diagnostic du Diabète.

### 3.5.1 La Définition du diabète[10]

Le diabète est une maladie chronique caractérisée par une concentration élevée de glucose(sucre) dans le sang, appelée hyperglycémie. Cela se produit parce que le corps ne produit pas suffisamment d'insuline (une hormone qui régule le taux de glucose dans le sang) ou parce que les cellules ne répondent pas correctement à l'insuline produite. Le diabète peut causer divers symptômes et complications à long terme s'il n'est pas traité correctement, notamment des problèmes de vision, de circulation sanguine, de reins et de système nerveux.

Le diabète est une maladie courante qui peut être traitée avec un régime alimentaire approprié, de l'exercice et, dans certains cas, des médicaments.

### 3.5.2 Les Types du diabète

On distingue principalement deux types de diabète : le diabète de type 1 qui touche environ 6% des diabétiques et le diabète de type 2 qui en touche 92 %. Les autres types de diabète concernent les 2 % restants

#### Diabète de type 1

Également appelé diabète insulino-dépendant ou juvénile, ce type de diabète survient lorsque le système immunitaire attaque et détruit les cellules productrices d'insuline dans le pancréas. Cela conduit à une production insuffisante d'insuline, et les personnes atteintes de diabète de type 1 ont besoin d'injections d'insuline pour contrôler leur glycémie. [36]

#### Diabète de type 2

Le diabète de type 2 (autrefois appelé diabète non insulino-dépendant ou diabète de la maturité) est le type le plus courant et survient lorsque le corps ne peut pas utiliser l'insuline efficacement ou ne produit pas suffisamment d'insuline pour répondre aux besoins du corps. Les personnes atteintes de diabète de type 2 peuvent souvent contrôler leur glycémie avec un régime alimentaire approprié, de l'exercice régulier et des médicaments, mais certaines peuvent également avoir besoin d'injections d'insuline. [37]

### 3.5.3 Symptômes du diabète

Les symptômes les plus fréquents du diabète sont : [37]

- Soif excessive et bouche sèche
- Besoin fréquent d'uriner
- Faim excessive
- Fatigue et faiblesse
- Vision floue
- Nausées / Vomissements
- Engourdissement ou picotement dans les pieds ou les mains
- Plaies qui guérissent mal
- Infections fréquentes et récurrentes.
- Problèmes de digestion
- Insuffisance rénale
- Crise cardiaque et accident vasculaire cérébral (AVC)

### 3.5.4 Diagnostic du diabète

Le diagnostic du diabète est établi grâce à une prise de sang qui dose le taux de sucre dans le sang. Pour les périodes de mesure nous avons : [21]

#### Taux de glycémie à jeun

Le diagnostic du diabète peut être confirmé par la mesure du taux de glucose dans le sang à jeun. Un taux de glucose à jeun égal ou supérieur à 7 mmol/L (126 mg/dL) à deux reprises est considéré comme diagnostique de diabète

#### Le test d'hyperglycémie provoquée (HGPO)

C'est conçu pour mesurer les niveaux de sucre dans le sang après l'ingestion de 75 g de glucides sous forme de boisson, et surveiller les niveaux de sucre dans le sang pendant 8 heures à jeun et toutes les 2 heures après avoir ingéré la solution. Un diagnostic de diabète est posé si les niveaux de sucre dans le sang sont supérieurs à 11,1 mmol/L, tandis qu'un diagnostic de prédiabète est posé si les niveaux de sucre dans le sang sont compris entre 7,8 mmol/L et 11,0 mmol/L.

### L'hémoglobine glyquée (HbA1c)

Le test de l'HbA1c est réalisé en prélevant un échantillon de sang dans un laboratoire d'analyse médicale et est recommandé à des intervalles réguliers, environ tous les 2 à 3 mois. . Il n'est pas nécessaire d'être à jeun pour la prise de sang.

### 3.5.5 Le raisonnement à partir de cas pour le diagnostic du Diabète

Le raisonnement à partir de cas (RàPC) est une méthode d'apprentissage automatique qui utilise des exemples de cas pour améliorer la prise de décision clinique. Dans le cas du diagnostic du diabète, le RàPC peut être utilisé pour aider les médecins à identifier les patients à risque de développer la maladie en analysant les symptômes, les antécédents médicaux et les résultats de tests. Le RàPC peut également être utilisé pour aider à diagnostiquer le diabète chez les patients présentant des symptômes ou des signes ambigus en comparant leur cas à des cas similaires dans une base de données de cas. Cela peut aider à améliorer la précision du diagnostic et à réduire les erreurs de diagnostic.

## 3.6 Conclusion

En conclusion, le raisonnement à base de cas est un outil efficace d'aide à la décision médicale pour le diagnostic et le traitement des maladies, y compris le diabète. Il permet aux médecins d'accéder rapidement à des informations pertinentes, d'identifier des schémas et des associations entre les symptômes et les conditions, et de prendre des décisions éclairées sur les soins à fournir aux patients. Bien que la technologie soit encore en développement, elle montre déjà un potentiel prometteur pour améliorer la qualité des soins de santé et la prise de décision clinique.

# Chapitre 4

## Le framework JColibri

### 4.1 Introduction

Dans ce chapitre, nous plongeons dans l'univers de Colibri Studio et de son composant central, jCOLIBRI. Colibri Studio est un environnement de développement puissant dédié à la création de systèmes de récupération basés sur les cas (CBR). Avec ses fonctionnalités avancées et ses assistants intuitifs, il simplifie considérablement le processus de conception et de développement de ces systèmes. Quant à jCOLIBRI, il s'agit d'une bibliothèque logicielle essentielle qui facilite l'implémentation et l'exécution des systèmes CBR. Au cours de ce chapitre, nous explorerons en détail les fonctionnalités offertes par Colibri Studio et jCOLIBRI, mettant en évidence leur importance dans la création de systèmes CBR performants et efficaces.

Dans ce chapitre, nous examinerons les assistants et les outils principales de Colibri studio .Nous explorerons également les principales fonctionnalités et l'architecture de jCOLIBRI,et ses interfaces principales.

### 4.2 COLIBRI Studio[28]

COLIBRI Studio représente la couche supérieure de la plateforme COLIBRI, assurant ainsi son implémentation. Il propose des outils de création visuelle indispensables à la génération de systèmes CBR, sans nécessiter de manipuler directement le code source. Cette solution s'appuie sur le framework jCOLIBRI et permet la composition des composants CBR. COLIBRI Studio est intégré à l'environnement de développement Eclipse, largement utilisé, ce qui lui confère l'avantage d'utiliser les fonctionnalités offertes pour la gestion des projets et du code source Java. Il facilite la création de projets CBR en configurant automatiquement les bibliothèques requises. Ces bibliothèques constituent une nouvelle version du framework jCOLIBRI qui est jCOLIBRI 3.[28]

### 4.3 Les assistants

COLIBRI Studio intègre plusieurs assistants qui simplifient le processus de conception de manière significative. Pour obtenir une description approfondie et un exemple pratique. Pour créer un nouveau projet CBR, vous pouvez utiliser les assistantes accessibles via "New menu"[28] :

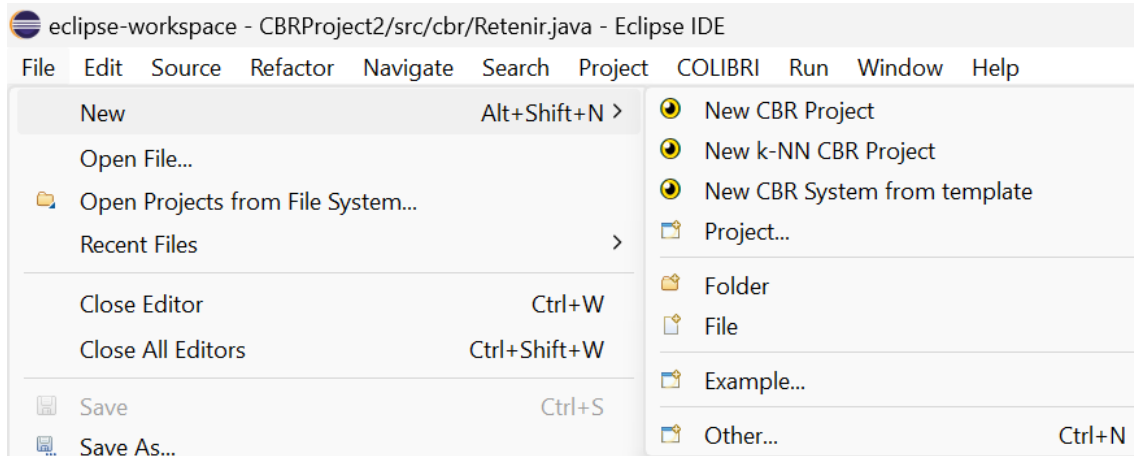


FIG. 4.1 : Assistants de COLIBRI studio

### 4.3.1 New CBR project wizard

Cet assistant facilite la création et la configuration d'un projet CBR vide. Les étapes comprennent la création du projet, la configuration des bibliothèques nécessaires, la sélection du package et de la classe Java pour l'implémentation du système CBR, ainsi que la définition de l'organisation de la base de cas en mémoire. Des options supplémentaires permettent de définir la structure des cas, les mesures de similarité et la persistance. Une fois terminé, le projet généré apparaîtra dans l'explorateur de packages d'Eclipse, avec un fichier Java contenant une base vide d'application CBR, prêt à être programmé ou généré avec l'outil System Editor.

### 4.3.2 Wizard New k-NN CBR Project

Le wizard "wizard New K-NN CBR project" est un outil qui permet de générer un système complet basé sur l'algorithme des k plus proches voisins (k-NN).

Ce assistant guide les utilisateurs à travers les étapes de configuration nécessaires pour créer un système k-NN fonctionnel. Les utilisateurs sont invités à définir la structure des cas, c'est-à-dire les caractéristiques ou attributs utilisés pour mesurer la similarité entre les cas. Ils peuvent également configurer les mesures de similarité spécifiques utilisées par l'algorithme k-NN.

En outre, les utilisateurs peuvent déterminer le mode de persistance des cas, c'est-à-dire comment les cas seront stockés et accessibles par le système k-NN. Cela peut inclure des options telles que le stockage en mémoire ou la persistance dans une base de données.

Une fois que toutes ces configurations sont définies, le wizard génère automatiquement un système k-NN entièrement fonctionnel. Les utilisateurs peuvent ensuite utiliser ce système pour effectuer des tâches telles que la classification, la prédiction ou la recommandation en se basant sur les cas existants dans le système.

### 4.3.3 New CBR system from template

Cet assistant permet de créer un nouveau projet CBR en sélectionnant un modèle à partir du catalogue, qui sera ensuite adapté pour générer le système CBR. Une fois la configuration du projet terminée, l'assistant demande le type de système CBR à générer (famille). Ensuite, un système de recommandation propose des modèles à l'utilisateur, associés à une implémentation réelle et une capture d'écran. Cette approche permet aux utilisateurs de tester le comportement du système CBR une fois implémenté, dans un domaine de test spécifique.

## 4.4 Les Outils principaux de COLIBRI studio

### 4.4.1 L'outil Case Designer

Le Case Designer est un outil essentiel pour définir la structure des cas dans jCOLIBRI. Les cas sont constitués de composants tels que : Description, Solution, Result et Justification, chacun contenant plusieurs attributs qui peuvent être simples ou composés. Chaque attribut est représenté par un Bean Java contenant son nom, son type et ses méthodes getter/setter. Il est obligatoire de définir un attribut simple comme identifiant (ID) pour chaque composant, car ces identifiants sont utilisés comme clés primaires par le connecteur de persistance.

La fonctionnalité des boutons de la barre d'outils du Case Designer consiste à faciliter la création, le chargement et l'enregistrement de nouvelles structures de cas au format XML, ainsi qu'à la génération des Beans Java correspondants.[28]

### 4.4.2 L'outil similarity

Dans COLIBRI Studio, l'outil de similarité est une fonctionnalité qui permet de définir la mesure de similarité utilisée dans un système CBR. Cette mesure de similarité est essentielle pour évaluer la similarité entre les cas dans la base de cas.

L'outil de similarité offre différentes options pour configurer la mesure de similarité en fonction des attributs utilisés dans le système CBR.

L'outil de similarité dans COLIBRI Studio facilite la personnalisation et l'ajustement des mesures de similarité selon les besoins du projet CBR, permettant ainsi une meilleure précision dans la recherche de cas similaires et dans les résultats du système CBR.

### 4.4.3 L'outil Case Base Selector

Après le chargement des cas depuis un support de persistance, cet outil permet de configurer l'organisation en mémoire de la base de cas. Il est également capable de générer le code source nécessaire à cette configuration. jCOLIBRI inclut les bases de cas suivantes :[29]

- `jcolibri.casebase.Lineal CaseBase` : une base de cas linéaire de base qui stocke les

cas dans une liste.

- `jcolibri.casebase.CachedLinealCaseBase` : une base de cas mise en cache qui ne persiste les cas que lors de la fermeture de l'application.
- `jcolibri.casebase.IDIndexedLinealCaseBase` : une extension de la base de cas linéaire qui conserve également un index des cas en utilisant leurs ID .
- `jcolibri.casebase.KDTreeCaseBase` : une base de cas qui stocke et organise les cas dans un arbre, ce qui permet une recherche efficace des cas similaires.

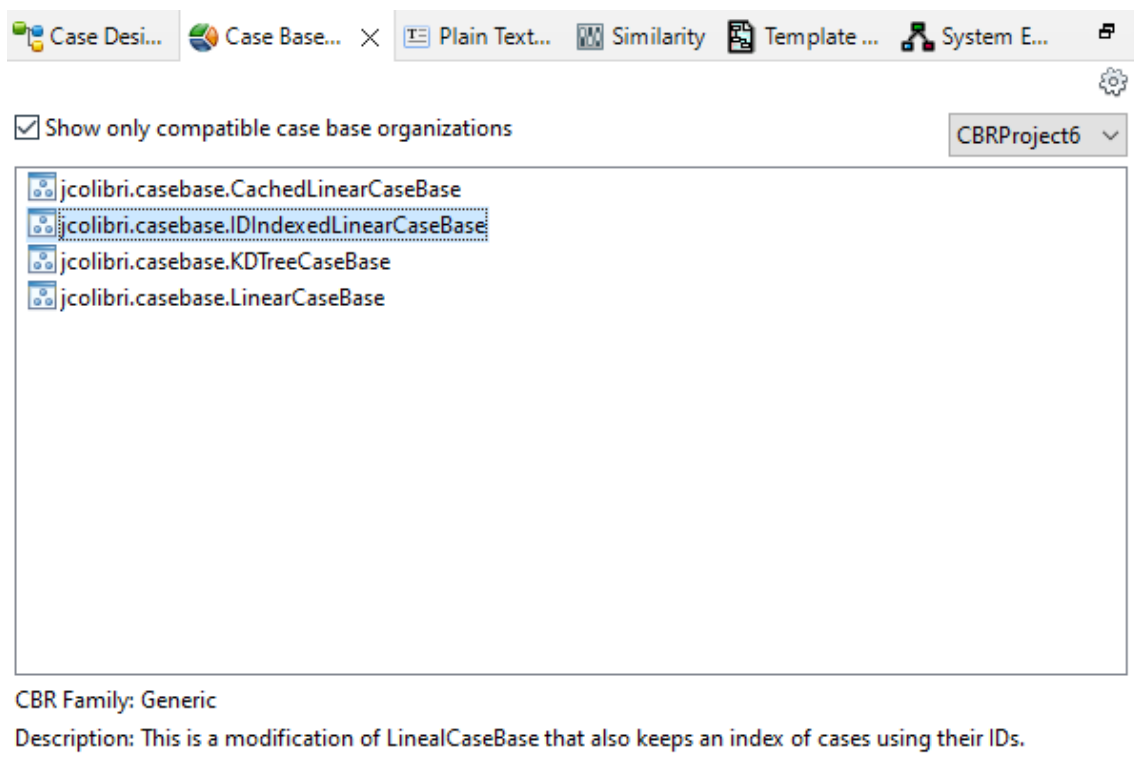


FIG. 4.2 : Capture d'écran de l'outil Case Selector

### 4.4.4 L'outil plain text connector configurator

Il est possible de configurer le connecteur Plain Text de jCOLIBRI en sélectionnant un fichier texte contenant les cas, puis en spécifiant le caractère de séparation des attributs. Pour ce faire, l'utilisateur peut utiliser cette outil, qui permet de choisir le fichier, de définir le caractère séparateur et d'effectuer la correspondance entre chaque colonne et un attribut du cas (dont la structure est préalablement définie dans l'outil Case Designer). Chaque cas est défini sur une ligne unique, avec les attributs séparés par le caractère de séparation désigné.[28]

### 4.5 jCOLIBRI Framework

Juan A. Recio-Garcia et son équipe de chercheurs en CBR ont conçu et entretiennent jCOLIBRI, un framework CBR Java libre et open source. Cet outil complet et efficient permet le développement de plusieurs types d'applications de RàPC, allant des systèmes textuels, structurés, de connaissances aux systèmes intensifs en données.

Il existe plusieurs outils disponibles pour construire un système De RàPC en dehors de jCOLIBRI, tels que Kate, ReCall, CBR-Works, ReMind, CBR\*Tools, CASPIAN, CAT-CBR, Indiana University Case Based Reasoning Framework (IUCBRF) et myCBR. Bien que IUCBRF soit techniquement plus simple et moins complexe que jCOLIBRI, ce projet étant obsolète et non pris en charge, il n'est pas recommandé de l'utiliser. Cependant, parmi ces outils, jCOLIBRI est le plus à jour, le plus pris en charge et dispose d'une communauté active de développeurs de systèmes de RàPC. La plupart des travaux liés aux systèmes de RàPC trouvés au cours de cette recherche ont été implémentés avec succès en utilisant jCOLIBRI.[39]

#### 4.5.1 jCOLIBRI version 3

La version 3 de jCOLIBRI est une nouvelle implémentation qui suit une architecture claire et divisée en deux couches distinctes : l'une orientée vers les développeurs (terminée) et l'autre orientée vers les concepteurs (travail futur). Ce nouveau design est un cadre complet en boîte blanche ouvert aux développeurs Java qui souhaitent inclure les fonctionnalités de jCOLIBRI dans leurs applications CBR.[8]

### 4.6 Architecture de jCOLIBRI3

La version 3 de jCOLIBRI adopte une approche de boîte blanche dans laquelle l'objectif principal est de permettre aux développeurs d'intégrer facilement jCOLIBRI dans leurs applications utilisant le RàPC. Cette architecture, destinée aux développeurs, fournit une base de conception pour les systèmes de RàPC qui est extensible et réutilisable. Elle offre également un ensemble de fonctions et de classes Java de base, ainsi que la possibilité pour les programmeurs d'inclure de nouvelles classes et fonctions pour répondre aux besoins spécifiques de leurs applications. [8]

L'architecture de la version 3 de jCOLIBRI propose de nouvelles fonctionnalités qui résolvent la plupart des problèmes identifiés dans la première version. Elle exploite les nouvelles possibilités offertes par les versions les plus récentes du langage Java. Le changement le plus important est la représentation des cas sous forme de Java Beans. La persistance des cas est gérée par le package Hibernate, une implémentation de Java Data Objects (JDO) qui permet de stocker automatiquement des Java Beans dans une base de données relationnelle, en utilisant une ou plusieurs tables. Les Java Beans et Hibernate sont des technologies clés de la plateforme Java 2 Enterprise Edition, qui est orientée vers les applications d'entreprise.[28]

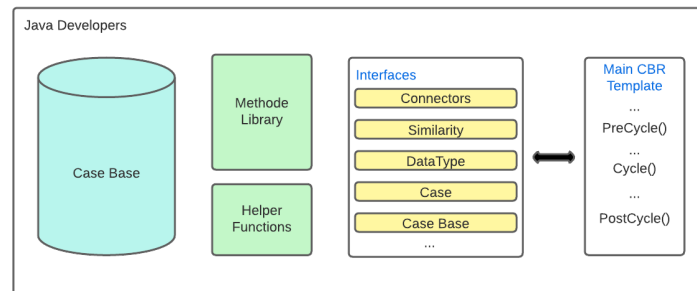


FIG. 4.3 : Architecture de jCOLIBRI 3. [8]

## 4.7 Les interfaces principales de jCOLIBRI[29]

### 4.7.1 Interface StandardCBRAApplication

L'interface localisée dans le package "cbrapplications" est considérée comme étant l'interface de configuration. Chaque nouvelle application qui utilise le RàPC doit implémenter `StandardCBRAApplication` ), car elle définit les méthodes standards nécessaires. En tout, il y a quatre méthodes distinctes définies dans cette interface :[29]

- **configure()** : cette fonction configure l'application en définissant la base de cas, les structures ,les connecteurs, et autres paramètres nécessaires.
- **precycle()** : Initialise l'application CBR consiste habituellement à charger la base de cas et à effectuer des calculs préalables sur des algorithmes coûteux (particulièrement utiles lors de l'utilisation de textes). Cette étape n'a lieu qu'une seule fois.
- **cycle(CBRQuery query)** :Exécute le cycle du RàPC. Cette étape est exécutée plusieurs fois.
- **postCycle()** : Cette fonction marque la fin de l'application du cycle RàPC. Elle a pour rôle de fermer les connecteurs à la base de données.

### 4.7.2 Interface CaseComponent

L'interface `jcolibri.cbrcore.CaseComponent` définit les composants d'un cas et permet aux développeurs de spécifier les attributs qui composent un cas. Elle garantit également que chaque cas possède un identifiant unique dans la base de données. Les composants de cas sont des Java Beans qui retournent un attribut unique pour chaque cas et peuvent inclure la description du problème, la solution au problème, le résultat de l'application de la solution et la justification de la solution.[29]

### 4.7.3 Interface CBRCaseBase

Pour que les méthodes CBR puissent accéder aux cas, la base de cas implémente également une interface commune. Ainsi, l'organisation et l'indexation choisies pour la base de cas n'affecteront pas la mise en œuvre des méthodes. L'interface "jcolibri.cbrcore.CBRCaseBase" définit l'organisation des cas en mémoire[29].

- **init(Connector connector)** : Cette fonction permet d'initialiser la base de cas en utilisant un connecteur pour la gestion de la persistance des données. En cas d'échec, une exception de type "InitializingException" est renvoyée.
- **Close()** : Effectue la désinitialisation de la base de cas
- **Collection<CBRCase> getCases** : Cette méthode permet de récupérer l'ensemble des cas disponibles dans cette base de cas.
- **Collection<CBRCase>getCases(CaseBaseFilter filter)** : Retourne les cas qui correspondent aux critères de recherche spécifiés par un filtre.
- **learnCases(Collection<CBRCase> cases)** : Ajoute une collection de nouveaux objets CBRCase à la base de cas actuelle.
- **forgetCases(Collection<CBRCase> cases)** : Supprime une collection d'objets CBRCase de la base de cas actuelle.

### 4.7.4 Interface Connector

Dans jCOLIBRI, la persistance est construite autour des connecteurs, qui constituent la première couche de communication entre le système CBR et le stockage physique. Les connecteurs sont des objets capables d'accéder et de récupérer des cas à partir du support physique, et de les transmettre de manière uniforme au système de RàPC. jCOLIBRI3 comprend les connecteurs suivants :

- Le connecteur "jcolibri.connectors.DataBaseConnector" gère la persistance des cas dans les bases de données en utilisant la bibliothèque Hibernate en interne.
- Le connecteur "jcolibri.connectors.PlainTextConnector" assure la gestion de la persistance des cas dans des fichiers texte.
- Le connecteur "jcolibri.connectors.OntologyConnector" utilise OntoBridge2 pour gérer la persistance des bases de cas stockées dans des ontologies.

Pour un connecteur, il est évident que l'interface doit inclure des méthodes permettant de lire la Base de Cas en mémoire et de la mettre à jour dans les supports de persistants. Dans jCOLIBRI3, une interface nommée "Connector" est fournie dans le package "jcolibri.cbrcore". Chaque connecteur doit obligatoirement implémenter les méthodes définies dans cette interface :

- **initFromXMLfile(java.net.URL file)** :Initialise le connecteur avec le fichier XML donné.
- **Close()** :Effectuer le nettoyage de toutes les ressources utilisées par le connecteur et mettre le service en suspension.
- **storeCases(Collection<CBRCASE>cases)** : Enregistre les classes spécifiées sur le support de stockage
- **DeleteCases(Collection<CBRCASE> cases)** :supprime les cas de la mémoire
- **Collection<CBRCASE> retrieveSomeCases(caseBaseFilter filter)** : récupère certains cas en fonction du filtre spécifié.

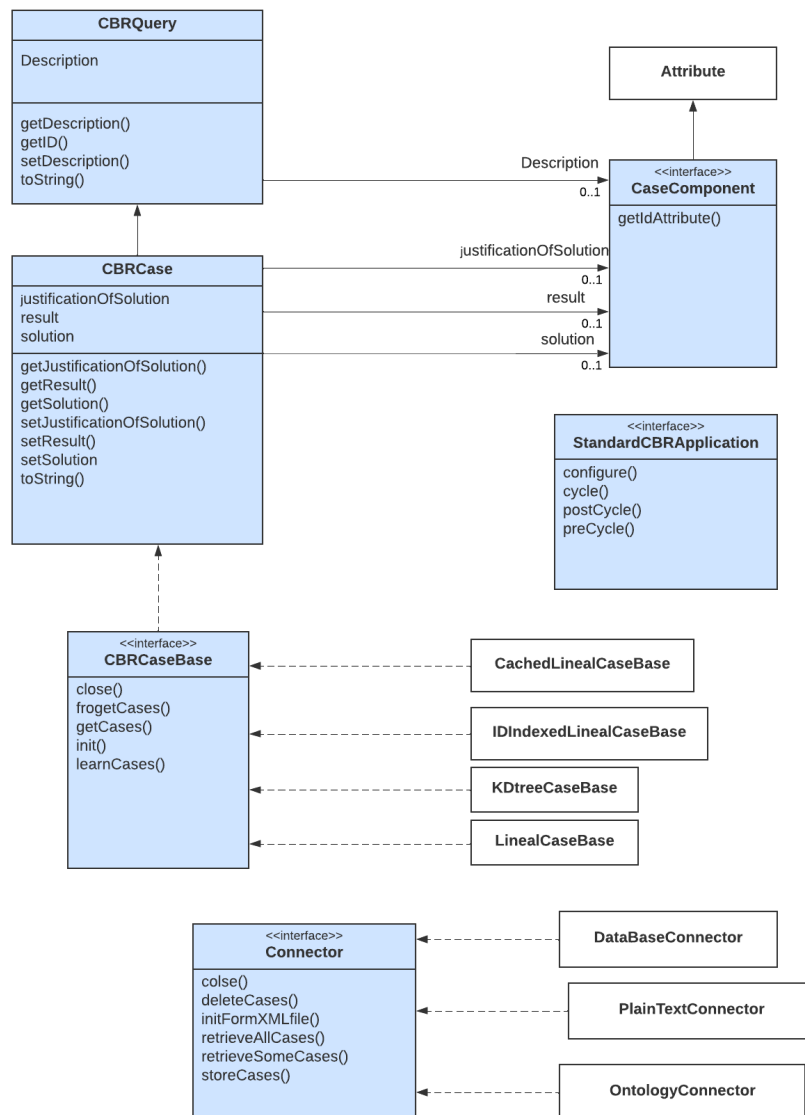


FIG. 4.4 : le diagramme UML de jCOLIBRI.[2]

### 4.8 Les fonctions de similarité

Dans la conception d'un système de RàPC à l'aide de JCOLIBRI, la configuration de similarité est une étape critique car elle permet de définir comment les cas doivent être comparés les uns aux autres pour identifier les cas les plus similaires à un nouveau cas donné. En outre, l'algorithme de récupération de cas le plus couramment utilisé dans ce contexte est le plus proche voisin.[8]

La configuration de la similarité des attributs dans JCOLIBRI varie en fonction de leur nature. Les attributs composés, par exemple, sont associés à une fonction de similarité globale, tandis que les attributs simples sont associés à des fonctions de similarité locale. Dans les deux cas, ces fonctions sont automatiquement générées à partir du framework jCOLIBRI.[28]

#### Les fonctions de similarité locales

Le framework jCOLIBRI offre aux développeurs une collection de fonctions de similarité locales pour calculer la similitude entre les attributs de deux cas. Chaque fonction est spécifiquement conçue pour un type de données particulier, tels que les chaînes de caractères, les textes, etc. Ces fonctions sont regroupées dans le package "similarity" et sont disponibles comme des fonctions de base, voici comment elles sont décrites :[8]

- **Fonctions de similarité entre les nombres** : Ces fonctions sont utilisées pour évaluer la distance entre les nombres , telles que :Threshold, Interval.
- **Fonctions de similarité entre les chaînes de caractères** : Ces fonctions sont utilisées pour évaluer la distance entre les chaînes de caractères telles que : Table, MaxString, EqualsStringIgnoreCase.
- **Fonctions de similarité pour les textes** : ces fonctions spécifiquement conçues pour le traitement de texte comme : CosineCoefficient, DiceCoefficient, JaccardCoefficient, LuceneTextSimilarity, OverlapCoefficient.
- **Fonction de similarité Equal** :est généralement utilisée pour les attributs catégoriels, où la similitude est déterminée par une correspondance exacte entre les deux valeurs.
- **Fonctions de similarité pour les ontologies** :sont employées pour mesurer la similarité dans le cadre d'une ontologie. Ainsi, lors de la conception d'un système utilisant une ontologie, il est possible de choisir l'une des fonctions suivantes : Ont-Cosine, OntDeep, OntDeepBasic, OntDetail.

#### Les fonctions de similarité globales

Dans JColibri, différentes fonctions de similitude peuvent être utilisées pour calculer la similarité globale entre deux cas , telles que :

- **Average** : représente une méthode pour calculer la similarité moyenne entre deux cas.

- **Euclidean** :représente une méthode pour calculer la distance euclidienne entre deux vecteurs de caractéristiques.
- **Minkowski** :est une fonction de similarité qui calcule la distance entre deux vecteurs de caractéristiques à l'aide de la formule de distance Minkowski. La distance Minkowski est une mesure de la distance entre deux points dans un espace à N dimensions.
- **Maximum** :est une fonction de similarité qui calcule la similarité maximale entre deux vecteurs de caractéristiques en prenant la valeur maximale pour chaque paire de caractéristiques correspondantes.

## 4.9 Conclusion

En conclusion, l'utilisation de Colibri Studio et jCOLIBRI pour la résolution de problèmes basée sur les cas se révèle extrêmement bénéfique. Ces outils offrent une large gamme de fonctionnalités avancées qui facilitent la gestion des connaissances, la récupération des cas, l'adaptation et la création d'interfaces utilisateur conviviales. L'exploration approfondie du code source démontre la possibilité d'implémenter de nouvelles fonctionnalités pour répondre à des besoins spécifiques. Les expériences menées avec jCOLIBRI ont mis en évidence les avantages de la résolution de problèmes basée sur les cas, notamment en termes de flexibilité et d'adaptabilité. En tant que développeurs, nous recommandons vivement l'utilisation de jCOLIBRI pour créer des applications intelligentes basées sur le CBR.

# Chapitre 5

## Conception

### 5.1 Introduction

Dans ce chapitre, nous abordons l'étape cruciale de la conception et de l'étude de cas pour notre mémoire portant sur le développement d'un système d'aide à la décision pour le diagnostic du diabète. Après avoir examiné en détail la méthodologie du Raisonnement à partir des Cas (RàPC) et analysé le framework de développement jCOLIBRI, notre objectif est d'appliquer cette méthodologie à la conception d'une solution permettant de diagnostiquer le diabète.

### 5.2 Etude de cas : diagnostic de diabète

Le diagnostic du diabète représente un défi majeur dans le domaine de la santé, ayant un impact significatif sur la santé des individus concernés. La précision du diagnostic nécessite une évaluation minutieuse des symptômes, des antécédents médicaux et des résultats de tests. Une prise en charge précoce et appropriée des patients diabétiques est essentielle pour réduire les complications et améliorer leur qualité de vie. Cependant, le processus de diagnostic rencontre plusieurs difficultés, notamment la complexité de l'analyse des données et la nécessité de traiter de vastes quantités d'informations. Dans cette étude, nous nous concentrons sur la conception d'un système d'aide à la décision pour le diagnostic du diabète. En utilisant des techniques de raisonnement à partir des cas et les avancées de l'intelligence artificielle, notre objectif est de fournir des recommandations diagnostiques précises et personnalisées. En améliorant l'efficacité et la précision du diagnostic, nous visons à offrir une prise en charge optimale aux patients diabétiques.

### 5.3 Présentation détaillée de notre Base de connaissance :

La base de connaissance utilisée dans notre étude provient du site Kaggle [12], comprennent des informations pertinentes pour le diagnostic du diabète.

La base de connaissance de prédiction du diabète est un ensemble de données médicales et démographiques recueillies auprès des patients, ainsi que de leur statut diabétique (positif ou négatif). Les données incluent des caractéristiques telles que l'âge, Body Mass Index (BMI), l'hypertension, heart Disease , le taux d'HbA1c et le taux de glucose sanguin. Cette base de données permet de construire des modèles d'apprentissage automatique pour prédire le diabète chez les patients en se basant sur leur historique médical et leurs informations démographiques. Cela peut être utile aux professionnels de la santé pour identifier les patients présentant un risque de développer le diabète et élaborer des plans de traitement personnalisés. De plus, la base de données peut être utilisée par les chercheurs pour explorer les liens entre différents facteurs médicaux et démographiques et la probabilité de développer le diabète.[12]

Voici une description détaillée des variables présentes dans notre ensemble de données :

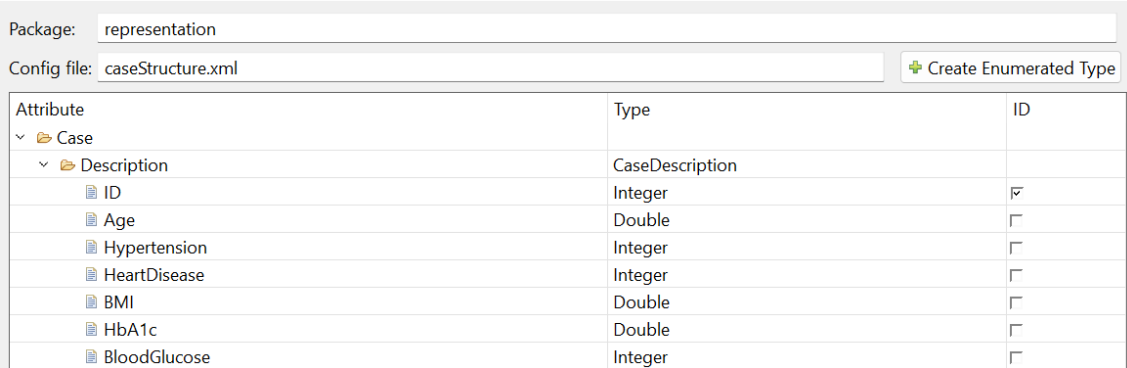
- Age : Cette variable représente l'âge du patient en années. L'âge peut être un facteur important dans le développement du diabète, car il est souvent associé à des changements physiologiques et à des risques accrus.
- Body Mass Index (BMI) : Le BMI est un indicateur utilisé pour évaluer le poids corporel d'une personne par rapport à sa taille. Il est calculé en divisant le poids en kilogrammes par le carré de la taille en mètres. Le BMI est un facteur clé dans l'évaluation du risque de diabète, car un indice élevé peut indiquer une obésité, qui est souvent liée à un risque accru de développer la maladie.
- Hypertension : Cette variable binaire indique la présence ou l'absence d'hypertension artérielle chez le patient. L'hypertension est une condition médicale courante qui peut être associée au développement du diabète.
- Heart Disease : Cette variable binaire indique si le patient présente une maladie cardiaque préexistante. Les maladies cardiaques sont souvent liées au diabète et peuvent augmenter le risque de complications chez les patients diabétiques.
- HbA1c : L'HbA1c est une mesure de la glycémie moyenne sur une période de plusieurs mois. Un niveau élevé d'HbA1c peut indiquer une mauvaise gestion du diabète ou une prédisposition au développement de la maladie.
- Blood Glucose : Cette variable représente le niveau de glucose sanguin dans le corps du patient. Les niveaux élevés de glucose peuvent être indicatifs de pré diabète ou de diabète.

### 5.4 Modélisation de la solution :

Après avoir recueilli les informations nécessaires à partir du site Kaggle et avoir discuté avec des médecins experts, nous sommes prêts à passer à la modélisation d'une solution dans le cadre d'un système d'aide à la décision pour la prise en charge du diagnostic du diabète. Notre approche s'appuiera principalement sur la méthodologie du Raisonnement à partir des Cas (RàPC)[29] .

#### 5.4.1 La représentation des données :

La représentation des données revêt une importance cruciale dans la mise en place d'un système de Raisonnement à partir des Cas (RàPC). En effet, cette représentation joue un rôle déterminant dans l'efficacité et la rapidité de la recherche des cas dans la base de données. Il est donc essentiel de sélectionner le type à stocker pour chaque cas et de déterminer la manière de les représenter. Comme il est montré dans la figure 5.1, la représentation des attributs dans notre système[2] .



Attribute	Type	ID
Case		
Description	CaseDescription	
ID	Integer	<input checked="" type="checkbox"/>
Age	Double	<input type="checkbox"/>
Hypertension	Integer	<input type="checkbox"/>
HeartDisease	Integer	<input type="checkbox"/>
BMI	Double	<input type="checkbox"/>
HbA1c	Double	<input type="checkbox"/>
BloodGlucose	Integer	<input type="checkbox"/>

FIG. 5.1 : représentation des cas

### 5.4.2 Gestion des connecteurs :

Les connecteurs sont des composants essentiels de COLIBRI Studio, permettant d'accéder et de récupérer de manière normalisée les cas à partir d'une persistance de cas spécifique. Ils assurent la gestion des différentes formes de stockage de cas en utilisant la structure de cas fournie. Les connecteurs agissent comme des interfaces entre le système CBR et les supports de stockage, en récupérant de manière cohérente les cas et en les transmettant au système CBR de manière uniforme. Grâce à cette flexibilité, COLIBRI Studio peut s'adapter à différents types de persistance de cas, du moment qu'un connecteur approprié est disponible. Cette fonctionnalité facilite également le chargement transparent des cas à partir de diverses sources de stockage, offrant ainsi aux utilisateurs une expérience fluide lorsqu'ils travaillent avec des données provenant de différentes sources. Le chercheur a utilisé un connecteur de fichiers texte pour la détection d'intrusion basée sur des cas, car les cas sont stockés au format texte. Ce connecteur assure la persistance des cas en spécifiant le chemin de la structure de cas et du fichier texte contenant les cas. Tous les attributs des cas sont associés et le connecteur est responsable de récupérer les données de la base de cas et de les renvoyer à l'interface graphique (GUI). Le connecteur lui-même est également sauvegardé au format XML, tout comme la structure de cas[2].

### 5.4.3 Configurations des structures de cas et des similarités.:

La définition de la structure de cas dans COLIBRI Studio est simple grâce à une fenêtre de gestion dédiée. Il est facile d'ajouter des attributs et de configurer les propriétés et les métadonnées. Les métadonnées incluent le poids, le type de données et la similarité. Les attributs les plus importants peuvent être déclarés avec un poids plus élevé. Dans la modélisation descriptive, la base de cas comprend 7 attributs de description et un de solution[2]. la figure 5.2 montre la description des attributs de cas avec le nom, le poids et la similarité locale.

Config file: similarityConfig.xml

Attribute	Similarity Function	Weight
Case		
Description	jcolibri.method.retrieve.NNretrieval.similarity.global.Average	
ID	Nothing	1.00
Age	jcolibri.method.retrieve.NNretrieval.similarity.local.Interval	0.50
Hypertension	jcolibri.method.retrieve.NNretrieval.similarity.local.Equal	0.50
HeartDisease	jcolibri.method.retrieve.NNretrieval.similarity.local.Equal	0.50
BMI	jcolibri.method.retrieve.NNretrieval.similarity.local.Interval	0.90
HbA1c	jcolibri.method.retrieve.NNretrieval.similarity.local.Interval	1.00
BloodGlucose	jcolibri.method.retrieve.NNretrieval.similarity.local.Interval	1.00

FIG. 5.2 : Configurations des structures de cas et des similarités

### Fonctions de similarité Locale :

Dans le cadre de notre système, nous sommes concentrés sur le développement de mesures de similarité efficaces pour les attributs des cas. Étant donné que le Framework jCOLIBRI ne propose que des fonctions de similarité simples, notre objectif était de trouver des mesures qui puissent répondre à nos besoins spécifiques. Nous avons choisi d'implémenter deux fonctions de similarité locale pour comparer les attributs des cas. Ces fonctions sont :

- Interval () : cette fonction est utilisée pour les attributs numériques lors de la récupération de voisins les plus proches. Cette mesure calcule la similarité en utilisant une mesure basée sur les intervalles de valeurs. L'idée principale derrière Interval est de comparer si deux valeurs numériques se situent dans un intervalle spécifique. La mesure de similarité est déterminée en fonction de la proximité de ces valeurs avec l'intervalle donné.
- Equal () : une fonction simple pour les attributs qui ont des valeurs fixes, si deux attributs sont égaux elle retourne 0, sinon elle retourne 1.

### Fonctions de similarité globale :

Une fois les mesures de similarité entre attributs calculées, la similarité globale calcule la distance entre deux cas. Elle utilise une fonction de calcul de similarité globale. Dans le cadre de ce projet, nous utilisons une mesure de similarité globale basée sur la technique du plus proche voisin puisqu'elle est la plus utilisée dans les systèmes basés sur le RàPC. Dans cet algorithme, la similarité entre le nouveau cas et le cas antérieur est calculée à travers la sommation de similarités entre les attributs (deux à deux) auxquels sont associés des poids. Cette similarité est la similarité globale Average [29]

L'algorithme est défini selon l'équation :

$$\text{sim}(A, B) = \sum_{i=1}^n w_i \text{sim}_i(a_i, b_i)$$

Où :  $n$  est le nombre d'attributs.

$w_i$  est le poids (évalué en fonction de l'importance) de l'attribut  $i$ .

$\text{sim}_i$  est la similarité locale calculée pour l'attribut  $i$ .

### 5.4.4 Creation de CBR application :

L'interface standard CBRApplication de jCOLIBRI est un élément essentiel pour développer une application de Raisonnement à partir de Cas avec jCOLIBRI. Cette interface définit les méthodes et les fonctionnalités nécessaires pour créer une application CBR complète[2].

```
public class CBRApplication implements StandardCBRApplication
// Méthode pour configurer l'application CBR
public void configure() throws ExecutionException;
// Méthode pour exécuter les étapes pré-cycle de l'application CBR
public CBRCaseBase preCycle() throws ExecutionException;
// Méthode pour exécuter les étapes du cycle de l'application CBR avec une requête
donnée
public void cycle(CBRQuery query) throws ExecutionException;
// Méthode pour exécuter les étapes post-cycle de l'application CBR
public void postCycle() throws ExecutionException;
```

FIG. 5.3 : la conception globale de notre système.

Après la génération de la fonction principale du code Java CBR, l'étape suivante pourrait consister à concevoir la structure du cas. Cette structure définit un composant de cas appelé CaseDescription pour la description du cas, ainsi qu'un autre composant appelé CaseSolution pour la solution. Les attributs d'identification (DescriptionID et SolutionID) sont requis par les composants de jCOLIBRI pour gérer les cas et servent de clés primaires pour l'indexation et le stockage de la base de cas[2].

```
package representation;
import jcolibri.cbrcore.Attribute;

public class CaseDescription implements CaseComponent {
    private java.lang.Integer ID;

    public java.lang.Integer getID()
    {
        return ID;
    }
    public void setID(java.lang.Integer ID0)
    {
        this.ID = ID0;
    }
    //rest of the code
```

FIG. 5.4 : le code de représentation de CasDescription.

```

package representation;

import jcolibri.cbrcore.Attribute;

public class CaseSolution implements CaseComponent {

    private java.lang.Integer Outcome;

    public java.lang.Integer getOutcome()
    {
        return Outcome;
    }
    public void setOutcome(java.lang.Integer Outcome7)
    {
        this.Outcome = Outcome7;
    }

    @Override
    public Attribute getIdAttribute()
    {
        return new Attribute("Outcome", this.getClass());
    }

    public String toString() {
        return "["+ Outcome +"]";
    }
}

```

FIG. 5.5 : le code de représentation de casSolution .

Une fois que la structure du cas est définie, l'étape suivante consiste à configurer la façon dont les cas sont chargés à partir du support de persistance. Configurez un connecteur qui charge les cas à partir de fichiers texte séparés par des virgules. Encore une fois, le champ "Config Palintext " contient le nom du fichier de configuration XML qui stockera les paramètres, et les connecteurs sont configurés à l'aide de fichiers de configuration XML[2].comme il est affiché dans la figure 5.6

Node	Content
?? xml	version="1.0" encoding="ISO-8859-1" standalone="no"
<ul style="list-style-type: none"> <li>▼ <input checked="" type="checkbox"/> TextFileConfiguration</li> <li> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> FilePath</li> <li><input checked="" type="checkbox"/> Delimiters</li> <li><input checked="" type="checkbox"/> DescriptionClassName</li> <li>&gt; <input checked="" type="checkbox"/> DescriptionMappings</li> <li><input checked="" type="checkbox"/> SolutionClassName</li> <li>&gt; <input checked="" type="checkbox"/> SolutionMappings</li> </ul> </li> </ul>	C:\Users\saram\Downloads\diabete0111.txt ; representation.CaseDescription representation.CaseSolution

FIG. 5.6 : Plain Text Connector Config.

### 5.4.5 Vue globale de notre système :

La figure 5.14 représente la conception globale de notre système :

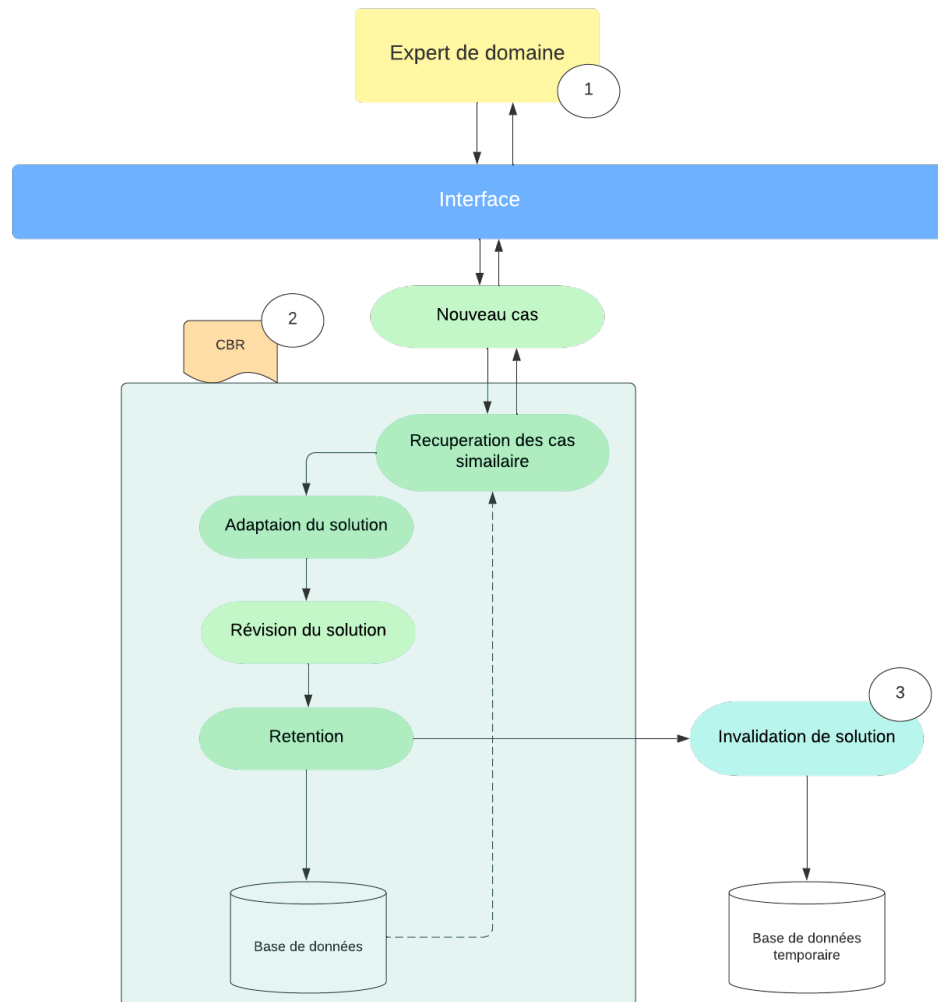


FIG. 5.7 : la conception globale de notre système.

L'explication de la conception globale de notre système :

1. L'expert de domaine utilise l'interface pour ajouter un nouveau cas de patient, en fournissant des informations telles que les symptômes, les résultats de tests pertinents. Cette étape permet d'intégrer de nouvelles données dans le système. la figure 5.8 explique le code java pour l'ajout de nouveau cas.

```
// Create a new CBR application instance
CBRApplication cbrApp = new CBRApplication();
// Configure the CBR application
cbrApp.configure();

CBRCaseBase casebase = cbrApp.preCycle();

// Create a new instance of the "Nouveau_Cas" class, passing the case base
Nouveau_Cas window = new Nouveau_Cas(casebase);
window.setVisible(true);

window.getRetrieve().addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        /**Get the attribute values from the interface**/
        double Age = Double.parseDouble(window.getageTextField().getText());
        int HyperTension = Integer.parseInt(window.getHypertensionTextField().getText());
        int HeartDisease = Integer.parseInt(window.getHeartDiseaseTextField().getText());
        double BMI = Double.parseDouble(window.getbmiTextField().getText());
        double HbA1c = Double.parseDouble(window.gethbA1cTextField().getText());
        int BloodGlucose = Integer.parseInt(window.getBloodGlucoseTextField().getText());
        CBRQuery query = new CBRQuery();
        CaseDescription caseDescription = new CaseDescription();
        query.setDescription(new CaseDescription());
            caseDescription.setAge(Age);
            caseDescription.setHypertension(HyperTension);
            caseDescription.setHeartDisease(HeartDisease);
            caseDescription.setBMI(BMI);
            caseDescription.setHbA1c(HbA1c);
            caseDescription.setBloodGlucose(BloodGlucose);
        query.setDescription(caseDescription);
    }
});
```

FIG. 5.8 : Code java pour l'ajout de nouveau cas.

2. Cette partie présente les étapes de la méthodologie de raisonnement à partir des cas dans notre système[34] :

- **Récupération de cas similaires** : L'étape de récupération permet d'obtenir les cas les plus similaires à partir d'une requête. La méthode principale de jCOLIBRI pour effectuer la récupération est la classe `jcolibri.method.retrieve.NNretrieval.NNScoringMethod`. Cette méthode effectue une comparaison de score numérique du voisin le plus proche en comparant les attributs. Elle utilise des fonctions de similarité globales pour comparer les attributs composés (Case-Components) et des fonctions de similarité locales pour comparer les attributs simples[2], donc une fois que le nouveau cas est ajouté, le système effectue une recherche dans la base de données existante pour trouver des cas similaires. Cette recherche se base sur les attributs et les caractéristiques du cas nouvellement ajouté. L'objectif est d'identifier des cas similaires qui peuvent fournir des informations pertinentes pour le diagnostic du patient en cours d'évaluation.

```
public void cycle(CBRQuery query) throws ExecutionException {  
  
    NNConfig simConfig = getSimilarityConfig();  
    Collection<RetrievalResult> eval = NNScoringMethod.evaluateSimilarity(casebase.getCases(), query, simConfig);  
    eval = SelectCases.selectTopKRR(eval, 5);  
  
    DefaultTableModel tableModel = new DefaultTableModel();  
    tableModel.addColumn("ID");  
    tableModel.addColumn("Case");  
    tableModel.addColumn("Similarity");  
    tableModel.addColumn("Solution");  
  
    // Add data to the table model  
    for (RetrievalResult nse : eval) {  
        Object[] rowData = {nse.get_case().getID(), nse.get_case().getDescription(), nse.getEval(),  
                            nse.get_case().getSolution() };  
        tableModel.addRow(rowData);  
    }  
  
    // Create the JTable  
    JTable table = new JTable(tableModel);  
    table.setBounds(50, 50, 400, 200);  
    table.getColumnModel().getColumn(0).setPreferredWidth(50); // ID column width  
    table.getColumnModel().getColumn(1).setPreferredWidth(270); // Case column width  
    table.getColumnModel().getColumn(2).setPreferredWidth(100); // Similarity column width  
    table.getColumnModel().getColumn(3).setPreferredWidth(70); // Solution column width  
  
    // Create a new JFrame  
    JFrame frame = new JFrame("Cas Retrouvés");  
  
    JOptionPane.showMessageDialog(null, new JScrollPane(table), "Cas Retrouvés", JOptionPane.PLAIN_MESSAGE, null);  
}
```

FIG. 5.9 : Code java " Récupération de cas similaires".

Il existe de nombreuses fonctions de similarité dans le package `jcolibri.method.retrieve.NNretrieval.similarity`. Pour cette étude, les fonctions de similarité "interval", "Equal" et "Average" sont utilisées pour mesurer la similarité.

```
private NNConfig getSimilarityConfig() {  
  
    NNConfig simConfig = new NNConfig();  
  
    simConfig.setDescriptionSimFunction(new jcolibri.method.retrieve.NNretrieval.similarity.global.Average());  
    // Add the DistanceSimilarityFunction to all attributes  
  
    Attribute attribute0 = new Attribute("Age", CaseDescription.class);  
    simConfig.addMapping(attribute0, new jcolibri.method.retrieve.NNretrieval.similarity.local.Interval(80));  
    simConfig.setWeight(attribute0, 0.5);  
  
    Attribute attribute1 = new Attribute("Hypertension", CaseDescription.class);  
    simConfig.addMapping(attribute1, new jcolibri.method.retrieve.NNretrieval.similarity.local.Equal());  
    simConfig.setWeight(attribute1, 0.5);  
  
    Attribute attribute2 = new Attribute("HeartDisease", CaseDescription.class);  
    simConfig.addMapping(attribute2, new jcolibri.method.retrieve.NNretrieval.similarity.local.Equal());  
    simConfig.setWeight(attribute2, 0.5);  
  
    Attribute attribute3 = new Attribute("BMI", CaseDescription.class);  
    simConfig.addMapping(attribute3, new jcolibri.method.retrieve.NNretrieval.similarity.local.Interval(95.7));  
    simConfig.setWeight(attribute3, 0.9);  
  
    Attribute attribute4 = new Attribute("HbA1c", CaseDescription.class);  
    simConfig.addMapping(attribute4, new jcolibri.method.retrieve.NNretrieval.similarity.local.Interval(9));  
    simConfig.setWeight(attribute4, 1.00);  
  
    Attribute attribute5 = new Attribute("BloodGlucose", CaseDescription.class);  
    simConfig.addMapping(attribute5, new jcolibri.method.retrieve.NNretrieval.similarity.local.Interval(300));  
    simConfig.setWeight(attribute5, 1.00);  
    return simConfig;  
}
```

FIG. 5.10 : Code java " les fonctions de similarity".

- **Adaptation** : Après la sélection des cas les plus proches, L'étape d'adaptation consiste à créer une copie du cas le plus proche sélectionné. Dans cette copie, les propriétés sont modifiées pour les adapter à la requête initiale. Par exemple, la description de la requête est copiée et attribuée à la description de la copie du cas, ce qui permet de mieux correspondre à la requête initiale. De plus, la solution du cas d'origine est également copiée et attribuée à la solution de la copie du cas, permettant de conserver la solution proposée dans le cas sélectionné, mais éventuellement de l'adapter ou de la modifier en fonction de la requête initiale. Cette étape d'adaptation vise à prendre en compte les informations spécifiques de la requête et à ajuster le cas sélectionné pour améliorer sa pertinence par rapport à la situation en question.

```
/****** Execute NN *****/
Collection<RetrievalResult> eval = NNScoringMethod.evaluateSimilarity(_caseBas.getCases(), query, simConfig);

/****** Select cases *****/
Collection<CBRCCase> selectedcases = SelectCases.selectTopK(eval, 1);
JOptionPane.showMessageDialog(null, "Le cas sélectionné"+ selectedcases);
System.out.println("Le cas sélectionné :");
System.out.println(selectedcases);
/****** Reuse *****/

ArrayList<CBRCCase> ress = new ArrayList<CBRCCase>();
for(CBRCCase orig: selectedcases)
{
    try {
        CBRCCase copy = orig.getClass().newInstance();
        copy.setDescription(copyCaseComponent(query.getDescription()));
        copy.setSolution(copyCaseComponent(orig.getSolution()));
        ress.add(copy);
    } catch (Exception E) {
        E.printStackTrace();
    }
}

Collection<CBRCCase> newcases = ress;

System.out.println("Combined cases");
for(jcolibri.cbrcore.CBRCCase c: newcases)
    System.out.println(c);
```

FIG. 5.11 : Code java " l'étape d'adaptation".

- **Révision** : Après l'adaptation, l'étape de révision intervient pour examiner attentivement le cas adapté. Dans cette étape, le système analyse les diagnostics, les traitements et les résultats associés au cas adapté. Sur la base de ces informations, le système ajuste le diagnostic initial pour améliorer sa précision. Dans le code fourni, la révision est réalisée en sélectionnant le meilleur cas adapté et en définissant de nouveaux identifiants pour les attributs composés. Cela permet au système de mettre à jour le cas révisé avec un nouvel identifiant.

```
/****** Revise *****/
// Lets store only the best case
CBRCCase bestCase = newcases.iterator().next();

// Define new ids for the compound attributes
HashMap<Attribute, Object> componentsKeys = new HashMap<Attribute, Object>();
componentsKeys.put(new Attribute("ID", CaseDescription.class), cle);
jcolibri.method.revise.DefineNewIdsMethod.defineNewIdsMethod(bestCase, componentsKeys);
JOptionPane.showMessageDialog(null, "ID"+ bestCase);
System.out.println("Cases with new Id");
System.out.println(bestCase);

/****** Retain *****/

StoreCasesMethod.storeCase(_caseBas, bestCase);
```

FIG. 5.12 : Code java " l'etapde de révision et retention".

- **Retention** : Si le cas nouvellement ajouté est jugé valide et pertinent, il peut être intégré à la base de connaissance. Cela permet d'enrichir la base de connaissances du système et de renforcer ses capacités de diagnostic à mesure que de nouvelles données sont intégrées.
3. Gestion des cas non valides : Si le cas est considéré comme non valide, atypique ou nécessitant une attention particulière, il va être ajouté à une base de données temporaire. Ces cas peuvent être examinés ultérieurement pour une analyse plus approfondie ou une prise de décision spécifique.

```
JButton btnEnregistrer = new JButton("Enregistrer Tempo ");
btnEnregistrer.setBounds(266, 467, 180, 41);
btnEnregistrer.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        String ID,age,Hypertension,HeartDisease,bmi,hbAlc,BloodGlucose;
        ID=IDtextField.getText();
        age =agetextField.getText();
        Hypertension=HypertensiontextField.getText();
        HeartDisease=HeartDiseasetextField.getText();
        bmi=bmitextField.getText();
        hbAlc=hbAlctextField.getText();
        BloodGlucose=BloodGlucosetextField.getText();

        String rq="insert into add_data values('"+ID+"','"+age+"','"+Hypertension+"',"
            + "'"+HeartDisease+"','"+bmi+"','"+hbAlc+"','"+BloodGlucose+"')";

        try{
            st=con.laConnection().createStatement();
            st.executeUpdate(rq);
            JOptionPane.showMessageDialog(null,"Enregistrement effectué avec succès!",
                null,JOptionPane.INFORMATION_MESSAGE);

        } catch (Exception e) {
            e.printStackTrace();
        }
        dispose();
        Retenir elv=new Retenir(casebase);
        elv.setVisible(true);
    }
});
```

FIG. 5.13 : Code java "l'enregistrement dans Data temporaire".

Une classe "Add data" a été ajoutée au diagramme de classe de JColibri pour gérer l'ajout de données temporaires. Cette classe comprend une méthode pour ajouter des cas à la structure de données temporaire. Elle permet de stocker temporairement les cas non valides ou nécessitant une attention particulière, pour une analyse ultérieure ou une prise de décision spécifique.



# Chapitre 6

## Implementation

### 6.1 Introduction

Ce chapitre explore la mise en œuvre pratique de notre système. L'objectif principal est de décrire en détail les différentes étapes de développement, les choix de conceptions, les technologies utilisées, ainsi que les résultats obtenus. Nous exploitons la plateforme JColibri en tirant parti de ses fonctionnalités avancées pour le traitement des cas de diabète.

### 6.2 Technologies utilisées

#### Eclipse

Eclipse<sup>1</sup> est un environnement de développement intégré multi-langage (IDE) complet. Il prend en charge divers langages de programmation tels que Java, C/C++, Python, et plus encore. Avec son éditeur de code avancé, ses outils de débogage et de compilation, ainsi que ses fonctionnalités de gestion de projets, Eclipse offre une plateforme puissante pour le développement logiciel. Il se distingue par sa flexibilité, sa vaste communauté d'utilisateurs et de développeurs, ainsi que par la possibilité d'ajouter de nouvelles fonctionnalités via des plugins.



FIG. 6.1 : Logo de Eclipse.

#### Colibri Studio

Colibri studio<sup>2</sup> est une version évoluée de la plateforme COLIBRI, offrant une interface visuelle conviviale pour générer des systèmes CBR sans manipulation directe du code source. Il repose sur le puissant framework jCOLIBRI, permettant une composition simple des composants CBR. Son intégration à l'environnement de développement Eclipse lui confère des fonctionnalités avancées de gestion de projets et de code source Java. Il facilite la création de "projets CBR" avec une configuration automatique des bibliothèques requises.



FIG. 6.2 : Logo de Colibri studio.

---

<sup>1</sup><https://eclipseide.org/> (visité le 25/05/2023).

<sup>2</sup><https://gaia.fdi.ucm.es/research/colibri/colibrstudio/> (visité le 25/05/2023).

### jCOLIBRI3

jCOLIBRI3<sup>3</sup> est le framework fondamental utilisé par COLIBRI Studio, fournissant une bibliothèque complète de classes et d'outils pour implémenter les fonctionnalités du système CBR. Avec des fonctionnalités avancées telles que la récupération, l'adaptation et la réutilisation des connaissances, jCOLIBRI constitue une base solide pour le développement de systèmes CBR en utilisant COLIBRI Studio.

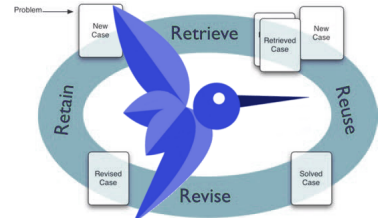


FIG. 6.3 : Logo de Jcolibri3.

### MySQL Workbench

MySQL<sup>4</sup> Workbench est une solution visuelle complète destinée aux architectes de bases de données, développeurs et administrateurs de bases de données (DBA). Cet outil unifié offre des fonctionnalités de modélisation de données, de développement SQL et d'administration avancées, couvrant la configuration du serveur, la gestion des utilisateurs, les sauvegardes et bien d'autres aspects. MySQL Workbench est disponible sur les plateformes Windows, Linux et Mac OS X, offrant une compatibilité étendue.



FIG. 6.4 : Logo de MySQL Workbench.

### Le langage Java

Java<sup>5</sup> est un langage de programmation orienté objet conçu pour la portabilité, la sécurité et la robustesse. Développé par Sun Microsystems (maintenant Oracle Corporation) en 1995, il est largement utilisé pour créer des applications web, des systèmes distribués et des applications mobiles. Grâce à sa machine virtuelle Java (JVM), il peut s'exécuter sur différentes plateformes. La syntaxe de Java est similaire à celle du langage C++, mais avec une approche simplifiée et axée sur la sécurité.



FIG. 6.5 : Logo de langage Java.

<sup>3</sup><https://gaia.fdi.ucm.es/research/colibri/jcolibri/> (visité le 25/05/2023).

<sup>4</sup><https://www.mysql.com/products/workbench/> (visité le 26/05/2023).

<sup>5</sup>[https://www.java.com/fr/download/help/whatis\\_java.html](https://www.java.com/fr/download/help/whatis_java.html) (visité le 26/05/2023).

### Kaggle

Kaggle <sup>6</sup> Une filiale de Google, est une plateforme en ligne qui réunit des data scientists et des ingénieurs en apprentissage automatique au sein d'une communauté dynamique. Elle offre aux utilisateurs la possibilité de trouver des ensembles de données pertinents pour la construction de modèles d'intelligence artificielle, de publier leurs propres ensembles de données, de collaborer avec d'autres experts du domaine et de participer à des compétitions visant à résoudre des défis en science des données. Lancé en 2010, Kaggle propose initialement des compétitions en apprentissage automatique et en science des données, et s'est développé pour devenir une plateforme d'éducation en science des données et en intelligence artificielle, offrant des ressources accessibles à tous via le partage de données et une infrastructure basée sur le cloud.



FIG. 6.6 : Logo de kaggle.

### UML

UML <sup>7</sup> est un langage de modélisation standard utilisé en génie logiciel. Il permet de représenter visuellement la structure, le comportement et les interactions des systèmes logiciels. Grâce à une variété de diagrammes, l'UML facilite la communication et la compréhension entre les différentes parties prenantes d'un projet. Il offre un moyen cohérent de visualiser et de documenter les concepts clés d'un système. L'UML est largement utilisé dans l'industrie du logiciel et est soutenu par de nombreux outils de modélisation.



FIG. 6.7 : Logo UML.

## 6.3 Installation de COLIBRI dans Eclipse

Nous avons débuté notre application CBR en installant COLIBRI Studio dans l'environnement Eclipse. Cette étape d'installation nous a permis d'intégrer COLIBRI Studio de manière fluide avec Eclipse, les figures suivantes illustrent les différentes étapes de l'installation du processus :[28]

---

<sup>6</sup><https://www.kaggle.com/> (visité le 27/05/2023).

<sup>7</sup><https://www.uml.org/> (visité le 27/05/2023).

La 1er étape, nous sélectionnons l'option appropriée pour installer un logiciel à partir d'une installation existante dans Eclipse. Cela nous permet de spécifier la source à partir de laquelle nous souhaitons importer le logiciel.

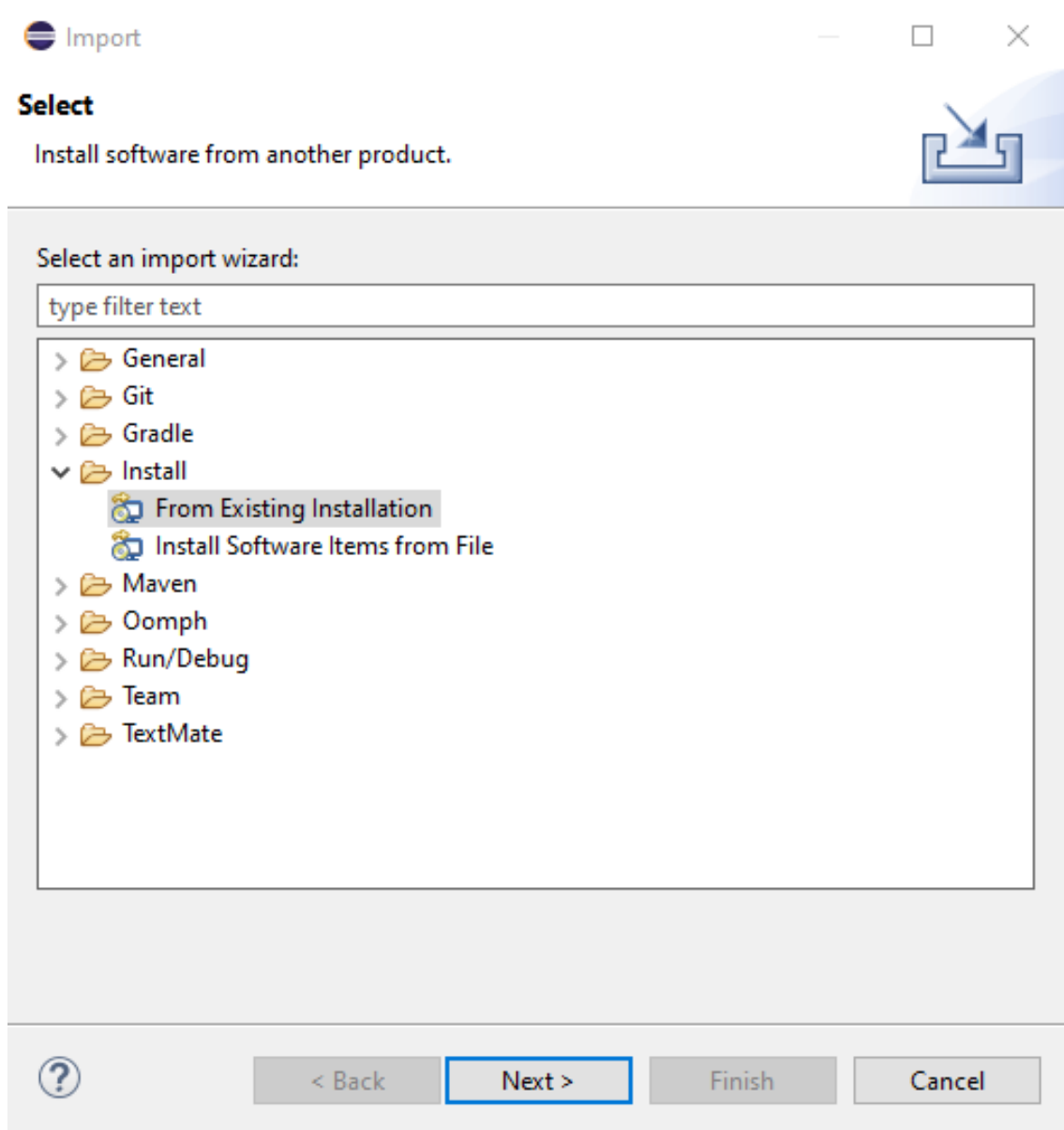


FIG. 6.8 : Étape 1 : Sélection de l'installation du logiciel.

La 2eme étape nous permet d'importer COLIBRI à partir d'une installation existante de l'application. Nous devons indiquer le chemin vers l'emplacement où COLIBRI est installé sur notre système.



La 4eme étape nous permet de consulter et d'accepter les licences associées aux composants de COLIBRI que nous sommes sur le point d'installer. Assurez-vous de lire attentivement les termes des licences avant de les accepter.

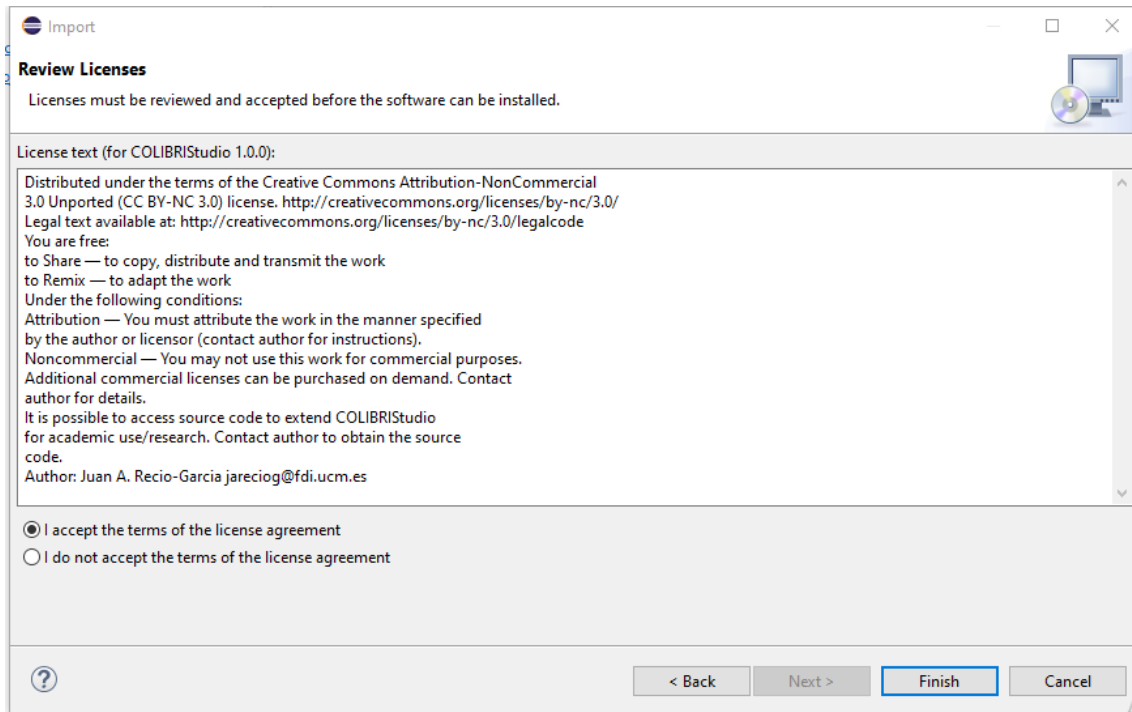


FIG. 6.11 : "Étape 4 : Vérification des licences".

Dans COLIBRI Studio, les outils et fonctionnalités sont regroupés dans une perspective Eclipse appelée "Perspective COLIBRI". Cette perspective est spécialement conçue pour offrir un environnement structuré et optimisé pour travailler avec COLIBRI Studio et ses fonctionnalités. En basculant vers la Perspective COLIBRI dans Eclipse, les utilisateurs peuvent accéder à un ensemble d'outils et de vues dédiés qui simplifient le développement et la gestion des systèmes CBR à l'aide de COLIBRI Studio. Cela permet une expérience de travail plus fluide et efficace en se concentrant sur les tâches spécifiques liées à COLIBRI Studio.[28]

La figure ci-dessous illustre le processus d'ouverture de la perspective COLIBRI dans Eclipse :

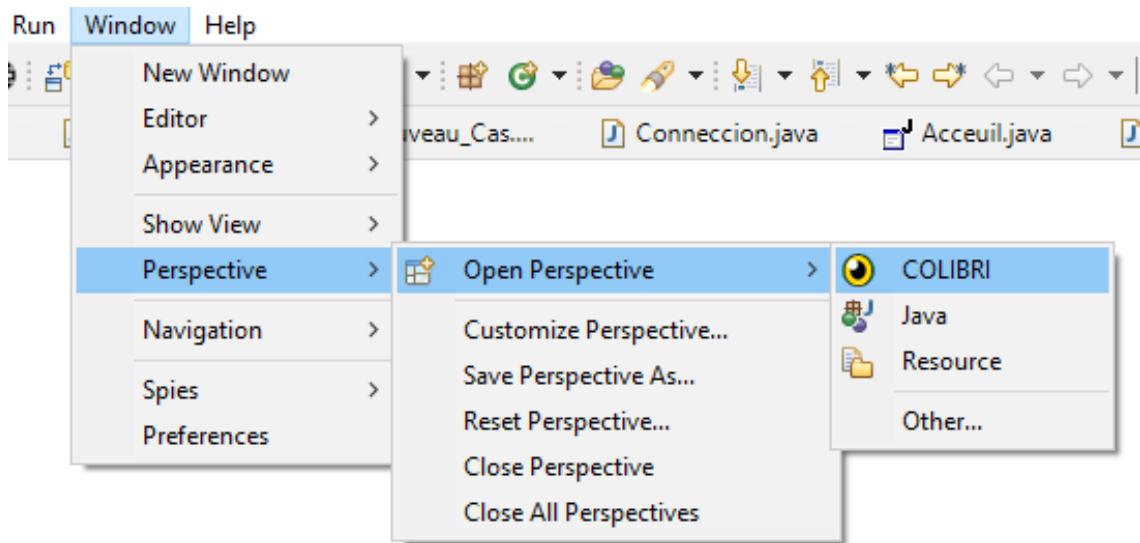


FIG. 6.12 : Ouverture de la perspective COLIBRI.

Après, on utilise l'algorithme KNN pour la réalisation de notre application CBR.

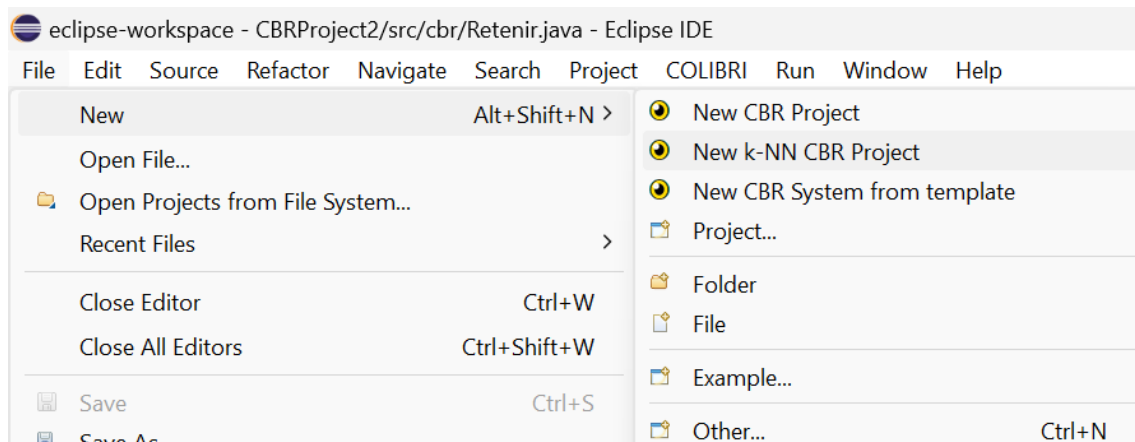


FIG. 6.13 : KNN projet

### 6.4 Présentation du notre système

Notre système a été conçu dans le but de simplifier le processus de prise de décision pour le traitement du diabète. Il comporte plusieurs étapes clés qui permettent d'optimiser cette démarche.

La première étape, appelée récupération, consiste à rappeler des informations pertinentes sur les symptômes et le diagnostic du patient. Cette récupération d'informations vise à fournir aux professionnels de la santé les éléments nécessaires pour prendre des décisions éclairées quant au traitement approprié.

Ensuite, notre système passe par les phases d'adaptation et de révision de la solution. L'adaptation se base sur les résultats obtenus lors de la récupération et permet d'ajuster la solution en fonction des besoins spécifiques de chaque nouveau cas. Une fois la solution adaptée, elle est révisée, validée et attribuée à un nouvel identifiant (ID) pour assurer une gestion efficace des cas.

Après la phase d'adaptation, notre système entre dans la phase de rétention. Durant cette étape, les nouveaux cas de patients sont stockés en vue d'une utilisation ultérieure dans le processus d'apprentissage et d'amélioration continue du système. Cette collecte de cas nous permet d'analyser les résultats obtenus, d'acquérir de nouvelles connaissances et de mettre à jour notre base de données. Cette approche contribue à améliorer nos décisions futures en offrant une approche plus précise et personnalisée pour le traitement et les soins des patients diabétiques. La phase de rétention joue un rôle fondamental dans l'évolution et l'efficacité de notre système d'aide à la décision.

### 6.5 Aperçu visuel de notre système

On présente les captures d'écran de notre application CBR pour vous permettre de découvrir son interface et ses fonctionnalités. Elles vous offrent un aperçu visuel des étapes de résolution de problèmes et des fonctionnalités clés disponibles.

#### 6.5.1 Accueil

Dès l'ouverture de notre application, vous découvrirez la page d'accueil présentant une interface claire. Au centre de cette interface, vous remarquerez un bouton intitulé "Nouveau Cas". Quand en cliquant sur ce bouton, vous ouvrirez une nouvelle page dédiée où vous pourrez saisir les symptômes de nouveau cas .

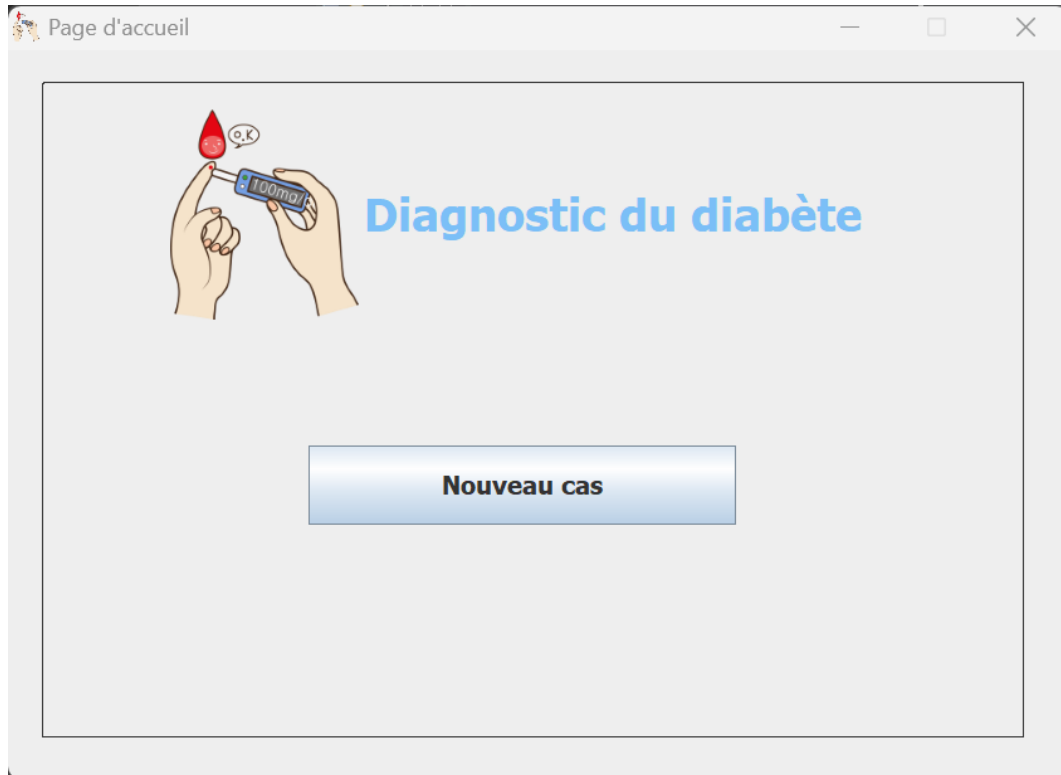


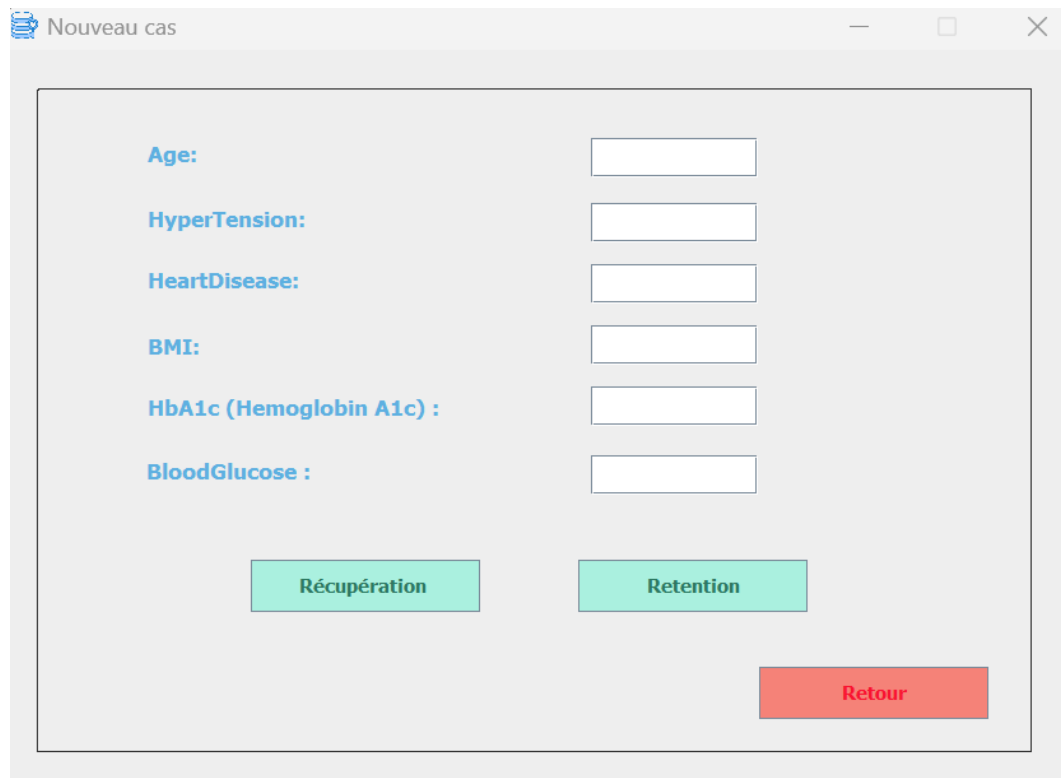
FIG. 6.14 : Interface accueil.

### 6.5.2 Interface Nouveau cas

Dans l'interface "Nouveau Cas", l'expert du domaine médical aura la possibilité de saisir les symptômes du nouveau cas dans les champs prévus à cet effet.

De plus, un bouton spécifique "Récupération" sera disponible pour récupérer les cas similaires. En cliquant sur ce bouton, notre application utilisera l'algorithme KNN pour rechercher les cas existants présentant des similarités avec le cas saisi. Cette fonctionnalité permettra de tirer parti des connaissances préexistantes afin de résoudre efficacement le problème en question.

Dans la page du nouveau cas, vous trouverez également un bouton nommé "Récupération", qui vous permettra d'accéder à une autre interface appelée "Retention".



The image shows a software window titled "Nouveau cas" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area is a light gray rectangle containing several labels in blue text, each followed by a white rectangular input field:

- Age:
- HyperTension:
- HeartDisease:
- BMI:
- HbA1c (Hemoglobin A1c) :
- BloodGlucose :

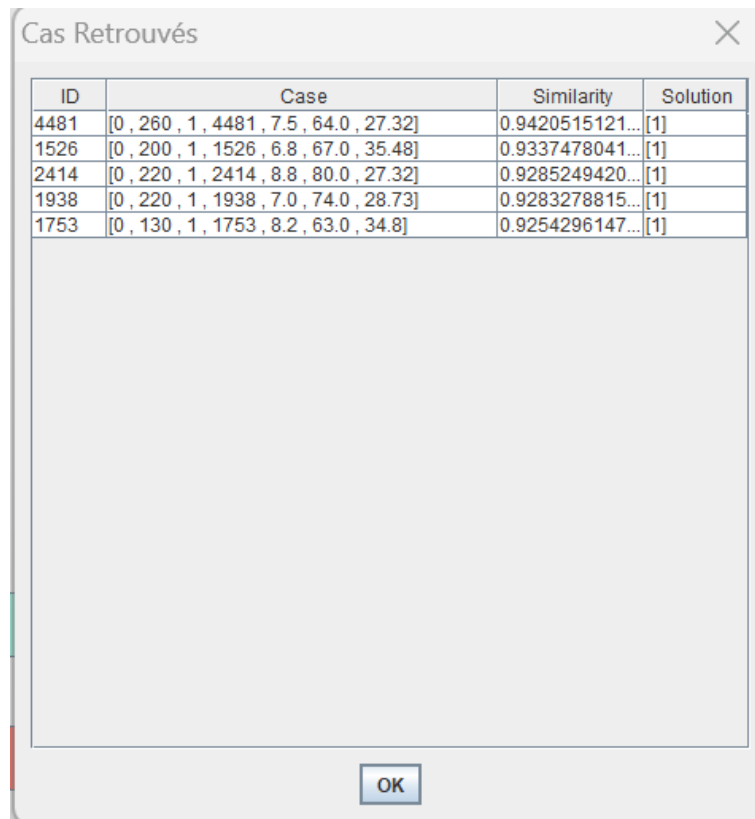
Below the input fields, there are three buttons:

- A light green button labeled "Récupération" on the left.
- A light green button labeled "Retention" in the center.
- A red button labeled "Retour" on the right.

FIG. 6.15 : Interface Nouveau Cas.

### 6.5.3 Interface Cas retrouvés

les 5 cas similaires récupérés seront affichés dans l'interface "Cas Retrouvés" sous forme de tableau. Chaque cas affiche sa description, son identifiant unique, sa mesure de similarité par rapport au cas de référence et le résultat correspondant. Ces informations sont présentées de manière claire dans une ligne du tableau dédiée à chaque cas.



ID	Case	Similarity	Solution
4481	[0, 260, 1, 4481, 7.5, 64.0, 27.32]	0.9420515121...	[1]
1526	[0, 200, 1, 1526, 6.8, 67.0, 35.48]	0.9337478041...	[1]
2414	[0, 220, 1, 2414, 8.8, 80.0, 27.32]	0.9285249420...	[1]
1938	[0, 220, 1, 1938, 7.0, 74.0, 28.73]	0.9283278815...	[1]
1753	[0, 130, 1, 1753, 8.2, 63.0, 34.8]	0.9254296147...	[1]

FIG. 6.16 : Interface Cas retrouvés.

### Exemple de l'étape de récupération

Dans l'exemple illustré dans la figure suivante, il s'agit de remplir les symptômes du diabète. Une fois les symptômes saisis, le système CBR recherchera les cas similaires dans la base de cas et les affichera dans le tableau correspondant.

Les informations fournies par les cas similaires aident le professionnel de la santé à prendre une décision éclairée en utilisant la similarité et les résultats des cas retrouvés. Cela permet d'obtenir une évaluation précise pour le nouveau cas.

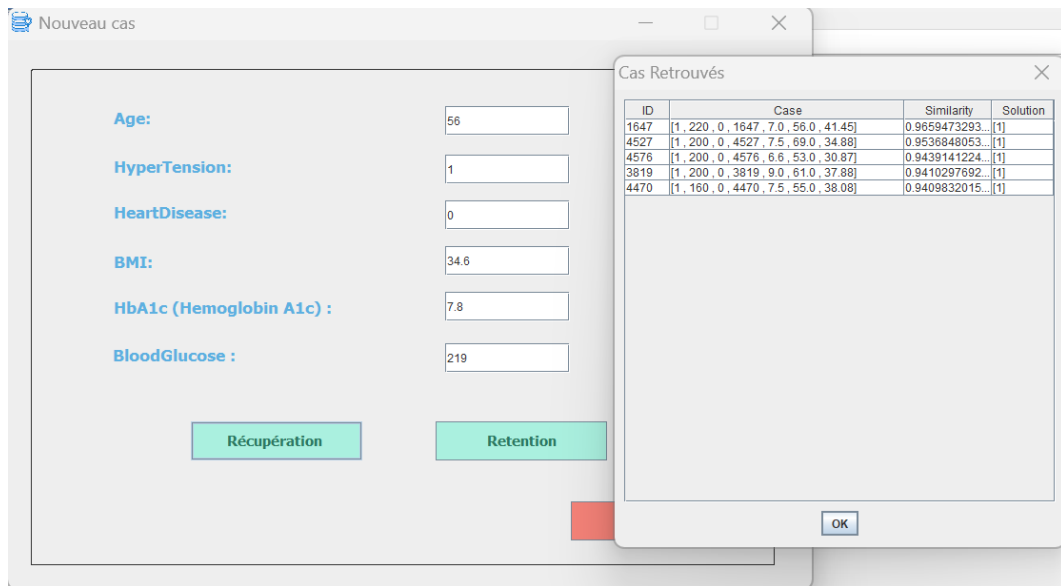


FIG. 6.17 : Exemple de nouveau cas et les cas retrouvés .

### 6.5.4 Interface Retenir

La figure suivante illustre l'interface "Retenir" qui facilite la phase de stockage pour enregistrer le nouveau cas après son adaptation et sa révision dans la base de cas. Cette interface comporte deux boutons essentiels : "Enregistrer BC" et "Enregistrer tempo".

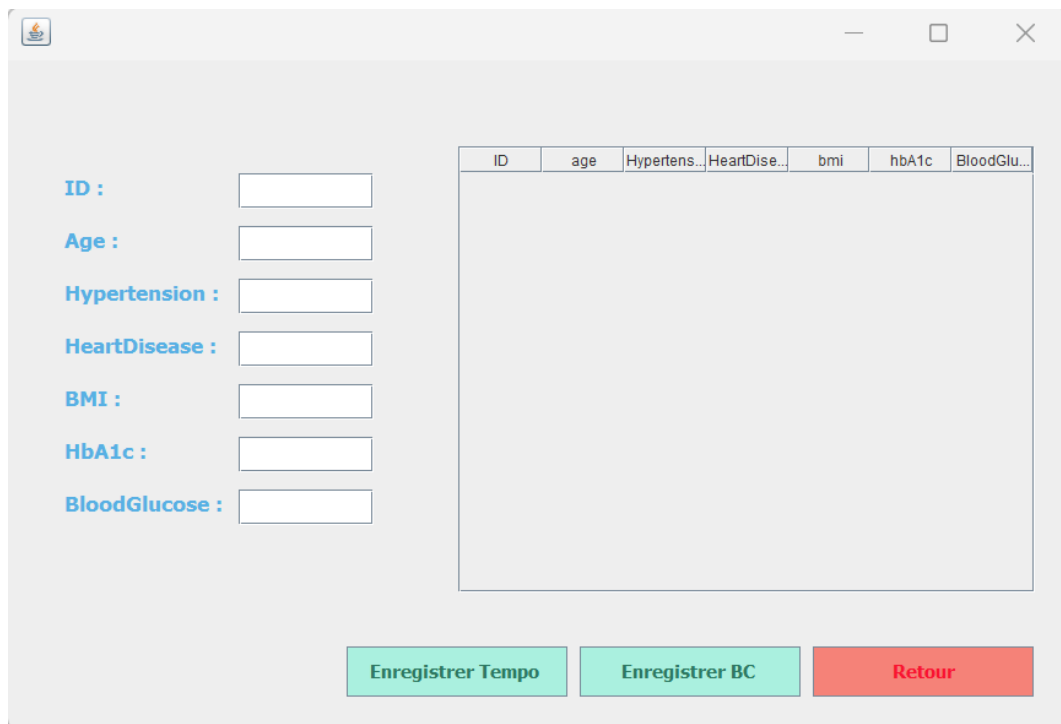


FIG. 6.18 : Interface retenir.

## Chapitre 6. Implementation

Lorsque le bouton "Enregistrer BC" est activé par l'expert du domaine dans notre système, le système effectue plusieurs étapes. Tout d'abord, il sélectionne le nouveau cas. Ensuite, il sélectionne les cinq cas les plus similaires retrouvés dans la phase de récupération. Parmi ces cinq cas, le système identifie le premier cas le plus proche du nouveau cas.

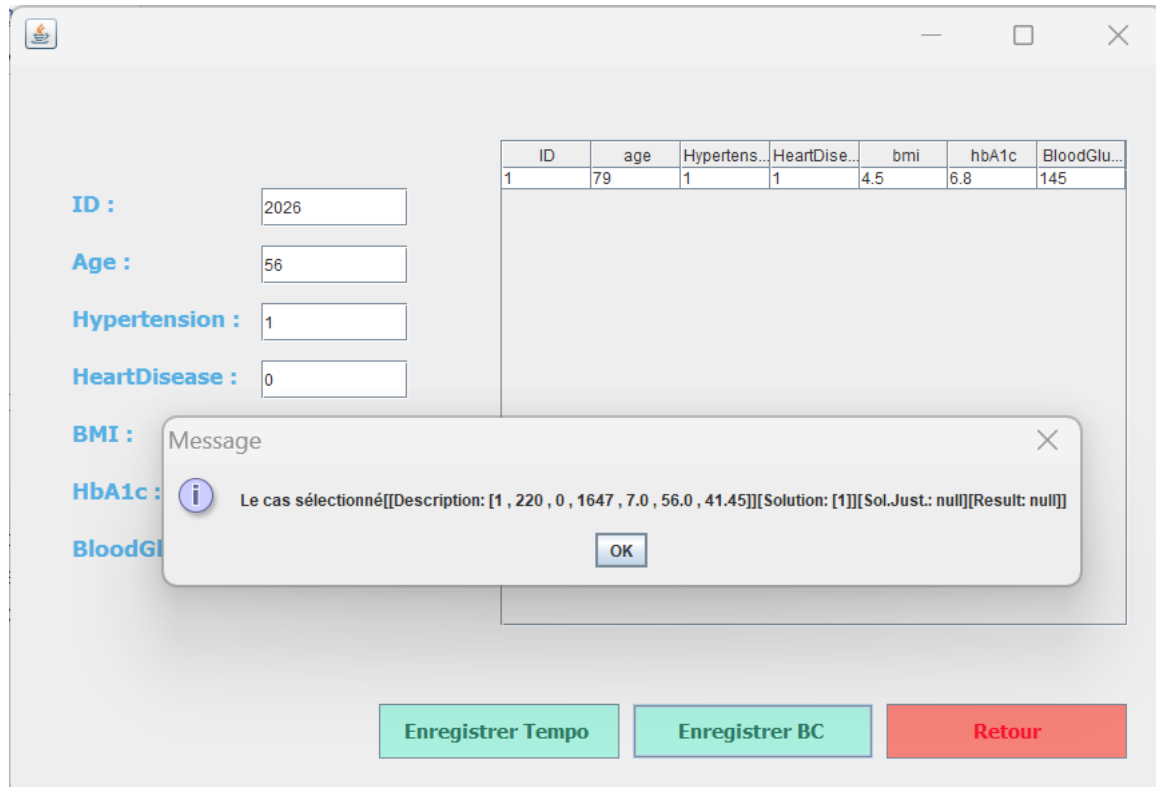


FIG. 6.19 : Le premier Cas sélectionné

Le système utilise ensuite ce premier cas comme référence pour adapter la solution du nouveau cas. Il effectue des ajustements basés sur les caractéristiques et les informations du premier cas similaire. Une fois la solution adaptée, le système procède à une révision pour s'assurer de sa pertinence et de sa précision.

Après la révision, le système attribue automatiquement un nouvel ID au cas enregistré. Cet ID est unique et permet d'identifier de manière spécifique ce cas dans la base de cas.

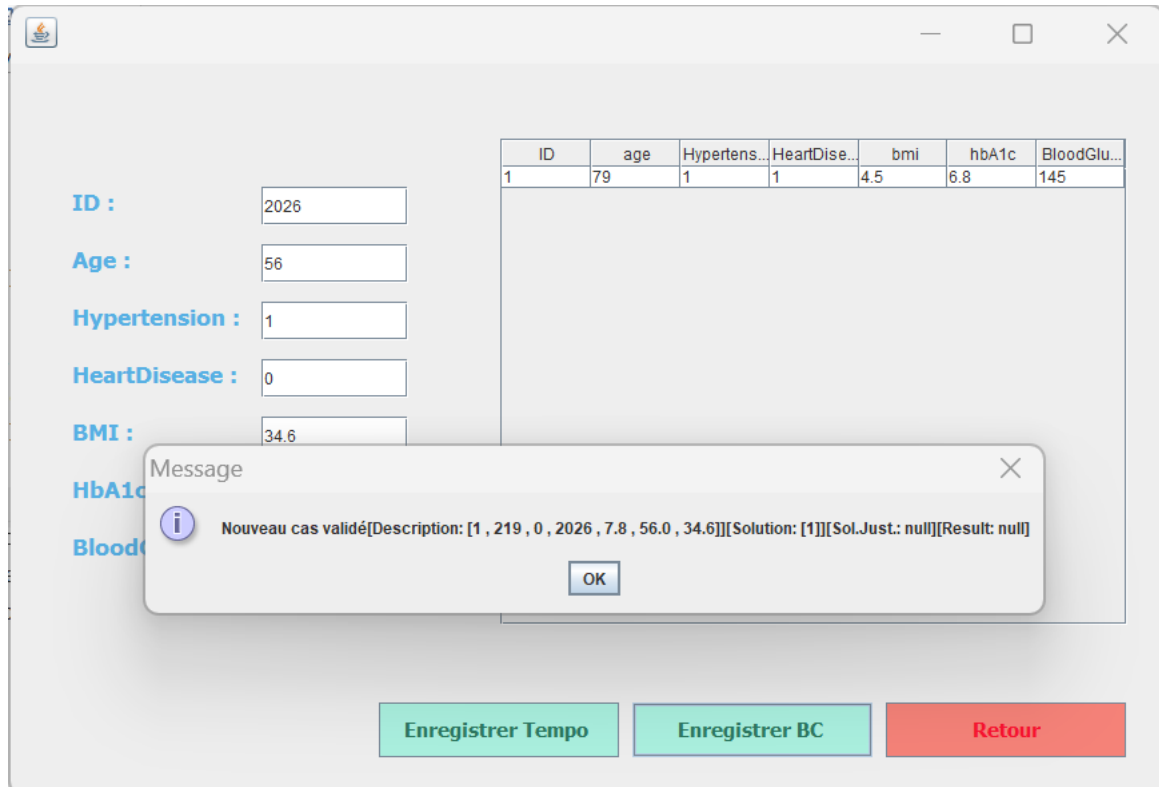


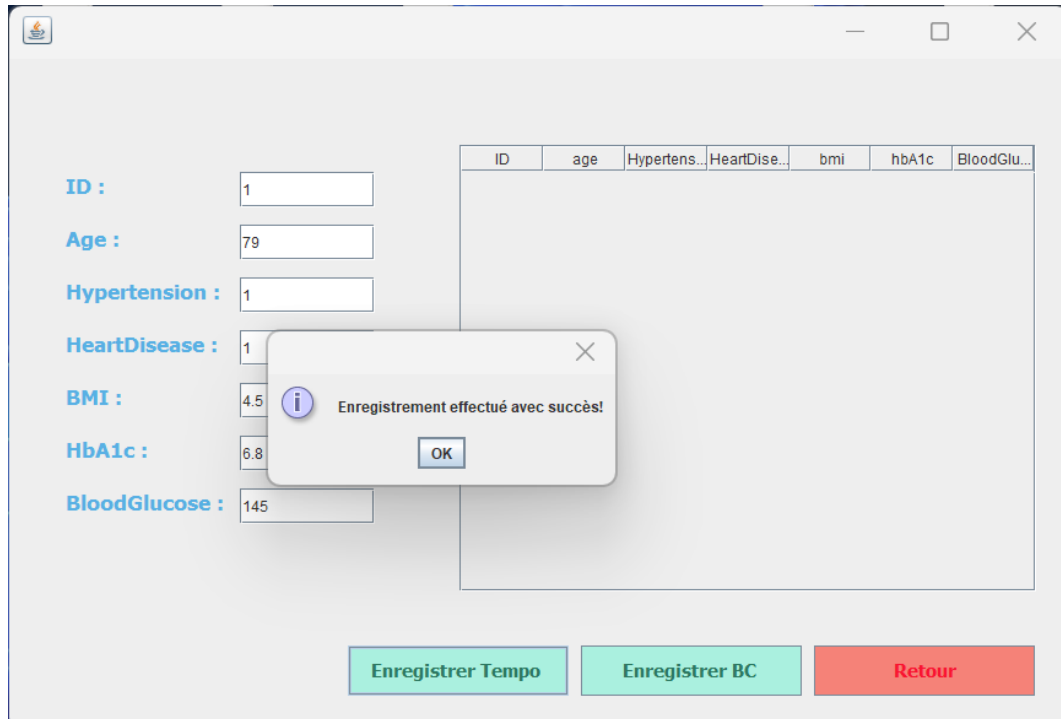
FIG. 6.20 : Le nouveau cas validé et identifier par son nouveau ID

Enfin, Le système enregistre de manière permanente le nouveau cas dans la base de données, l'incorporant de façon définitive. Cette intégration permet d'utiliser le cas pour l'apprentissage continu et les prises de décisions futures, garantissant ainsi une référence fiable et accessible en tout temps.

Cette approche permet d'améliorer progressivement le système CBR en enrichissant la base de cas avec de nouvelles expériences et connaissances acquises, renforçant ainsi son efficacité et sa pertinence.

Le bouton "Enregistrer tempo" intégrées dans cette interface offre la possibilité d'enregistrer le cas de manière temporaire.

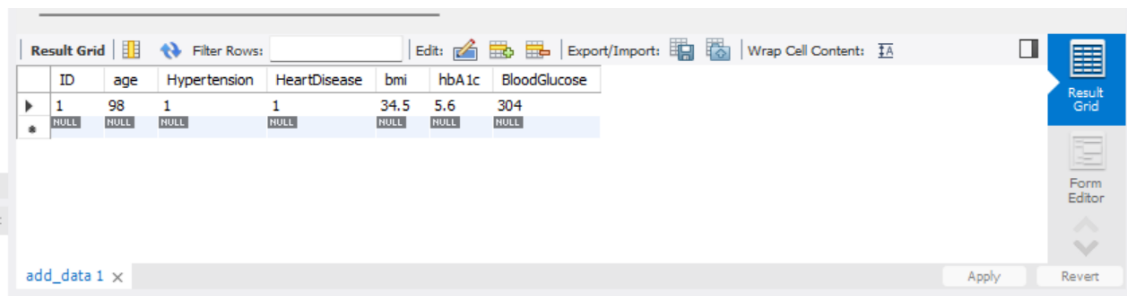
Si l'expert médical détecte des incertitudes concernant les résultats des cas similaires retrouvés dès le début pour le nouveau cas, il peut stocker ce cas dans une base de données temporaire. Cela permet de garder une trace de ces cas en suspens et de les revoir ultérieurement pour une évaluation plus approfondie ou une consultation supplémentaire avec d'autres experts. Cette base de données temporaire offre une flexibilité pour traiter les cas complexes et prendre des décisions plus informées à l'avenir.



The screenshot shows a web application interface for recording a new case. On the left, there are input fields for various attributes: ID (1), Age (79), Hypertension (1), HeartDisease (1), BMI (4.5), HbA1c (6.8), and BloodGlucose (145). A modal dialog box is open in the center, displaying the message "Enregistrement effectué avec succès!" (Registration successful!) and an "OK" button. At the bottom of the form, there are three buttons: "Enregistrer Tempo" (green), "Enregistrer BC" (green), and "Retour" (red). In the background, a table header is visible with columns: ID, age, Hypertens..., HeartDise..., bmi, hbA1c, BloodGlu...

FIG. 6.21 : Enregistrement de nouveau cas dans la base temporaire.

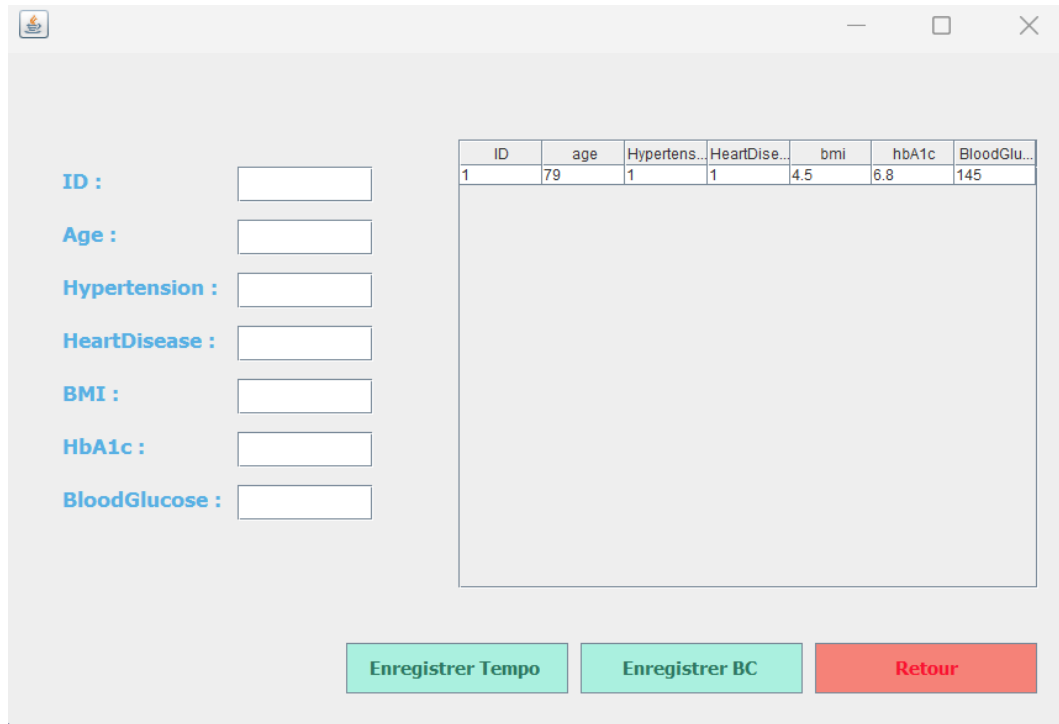
Les informations de cette base de données temporaire sont conservées dans une base MySQL et sont présentées dans un tableau compact dans l'interface "Retenir". Les deux figures ci-dessous fournissent une représentation visuelle de l'explication précédente.



The screenshot shows a MySQL data grid interface. The table has the following columns: ID, age, Hypertension, HeartDisease, bmi, hbA1c, and BloodGlucose. The first row contains the values: 1, 98, 1, 1, 34.5, 5.6, 304. The second row contains NULL values. The interface includes a toolbar with options like Filter Rows, Edit, Export/Import, and Wrap Cell Content. The bottom of the window shows the text "add\_data 1" and buttons for "Apply" and "Revert".

ID	age	Hypertension	HeartDisease	bmi	hbA1c	BloodGlucose
1	98	1	1	34.5	5.6	304
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIG. 6.22 : Enregistrement de data temporaire en MySQL .



The screenshot shows a web application interface with a form on the left and a data table on the right. The form contains input fields for ID, Age, Hypertension, HeartDisease, BMI, HbA1c, and BloodGlucose. The table displays a single row of data with the following values: ID: 1, age: 79, Hypertens.: 1, HeartDise.: 1, bmi: 4.5, hbA1c: 6.8, BloodGlu...: 145. At the bottom of the interface, there are three buttons: 'Enregistrer Tempo' (green), 'Enregistrer BC' (teal), and 'Retour' (red).

ID	age	Hypertens..	HeartDise...	bmi	hbA1c	BloodGlu...
1	79	1	1	4.5	6.8	145

FIG. 6.23 : L'affichage de data temporaire dans le tableau .

## 6.6 Test et évaluation du système

La mise en œuvre d'une évaluation et d'un test rigoureux du système est essentielle pour s'assurer de la qualité, de la fiabilité et de l'adéquation . Ces démarches permettent de quantifier les performances du système et de vérifier s'il répond aux attentes initiales et aux critères de réussite.

Des cas de test rigoureusement conçus sont utilisés pour évaluer les performances du système. Les résultats obtenus sont comparés aux évaluations des experts pour déterminer si le système produit des résultats cohérents et pertinents. Cette approche d'évaluation et de validation assure la qualité et la fiabilité du système, permettant d'identifier les ajustements nécessaires pour améliorer ses performances.[2]

### 6.6.1 Évaluation et test de la performance du notre système CBRDD

L'évaluation de la performance du CBRDD est essentielle pour vérifier s'il répond aux exigences de précision fixées. Cela permet de s'assurer que le système a été construit de manière adéquate. Pour cela, différentes mesures telles que la précision, le rappel et la F-mesure sont utilisées pour évaluer les performances du système. Ces critères permettent de quantifier précisément la précision du CBRDB et de déterminer dans quelle mesure il satisfait aux critères établis

## Chapitre 6. Implementation

---

Nous effectuons une analyse statistique sur un ensemble de 50 cas de diabete. Nous évaluons les performances du CBRDD en calculant les mesures de rappel et de précision ,et en utilisant un intervalle de degré de similarité de  $[1.0, 0.87]$ , seuls les cas ayant un score de similarité supérieur à 87% sont considérés comme pertinents.

Le calcul du rappel a été effectué en comparant le nombre d'exemples pertinents récupérés par le système avec les cas pertinents sélectionnés par les experts du domaine. Cette mesure permet d'évaluer la capacité du système à retrouver tous les cas pertinents[34]

La précision est calculée en divisant le nombre de cas pertinents récupérés par le nombre total de cas récupérés par le système. Le nombre total de cas récupérés comprend à la fois les cas pertinents et les cas non pertinents dans l'intervalle de seuil spécifié de  $[1.0, 0.87]$ . Cela permet d'évaluer la proportion de cas pertinents parmi tous les cas récupérés.[34]

Cas de test	Cas Choisis par l'expert	Cas choisis par le système	Cas totales par le système	Reappel	Précision	F-mesures
Cas 1943	13	11	16	0.84	0.68	0.75
Cas 1275	20	18	21	0.9	0.85	0.87
Cas 2597	6	6	6	1	1	1
Cas 1023	14	12	24	0.85	0.5	0.62
Cas 4496	16	13	17	0.81	0.76	0.78
Moyenne				0.88	0.75	0.80

TAB. 6.1 : Résultats de l'évaluation de notre système CBRDD

Ce tableau fournit une évaluation des performances de notre système CBRDD .Voici une analyse des résultats :

- Pour chaque cas de test, nous comparons le nombre de cas pertinents choisis par l'expert du domaine avec le nombre de cas choisis par notre système. Le nombre total de cas choisis par notre système est également indiqué.
- Le rappel mesure la capacité de notre système à récupérer les cas pertinents. Il est calculé en divisant le nombre de cas pertinents récupérés par le système par le nombre de cas pertinents choisis par l'expert. Dans notre étude, les valeurs de rappel varient de 0,81 à 1, ce qui indique que notre système a réussi à récupérer la majorité des cas pertinents dans la plupart des cas de test.

- La précision évalue la proportion de cas pertinents parmi tous les cas récupérés par notre système. Les valeurs de précision varient de 0,5 à 1, ce qui indique que notre système fournit des résultats précis dans certains cas, mais présente une précision plus faible dans d'autres cas.
- La F-mesure est une mesure qui combine le rappel et la précision pour donner une estimation globale de la performance du système. Les valeurs de F-mesure varient de 0,62 à 1, ce qui montre une performance variable du système en fonction des cas de test.

En analysant les résultats moyens, nous constatons un rappel moyen de 0,88, une précision moyenne de 0,75 et une F-mesure moyenne de 0,80. Ces résultats indiquent une performance globalement satisfaisante de notre système, mais il y a encore des possibilités d'amélioration pour augmenter la précision et la cohérence des résultats.

Il est important de noter que ces résultats doivent être interprétés en tenant compte du contexte spécifique de notre système CBRDD et des cas de test sélectionnés. Des ajustements et des améliorations peuvent être nécessaires pour renforcer la précision et l'efficacité de notre système dans des scénarios réels.

### 6.7 Conclusion

En somme, ce chapitre a présenté notre système sophistiqué d'aide à la prise de décision pour le diabète, offrant aux médecins une précieuse assistance dans leurs processus décisionnels. Grâce à ses interfaces conviviales, notre application permet une exploration approfondie des cas similaires, une adaptation minutieuse des nouveaux cas et une rétention organisée dans une base de données dédiée. En combinant l'intelligence artificielle et l'expertise médicale, notre système vise à réduire les incertitudes et à améliorer la qualité des décisions médicales, contribuant ainsi à l'amélioration des soins aux patients diabétiques.

# Conclusion générale

Notre mémoire s'est concentré sur le domaine de l'aide à la décision médicale et surtout sur l'intégration du raisonnement à partir de cas dans le diagnostic du diabète. Notre objectif était de répondre à la problématique de l'amélioration de la précision et de l'efficacité des systèmes d'aide à la décision médicale.

Nous avons démontré que le raisonnement à partir de cas offre une approche prometteuse pour le diagnostic du diabète. En réutilisant les expériences passées et en s'appuyant sur une base de connaissances étendue, ce type de système peut fournir des informations précises, aidant ainsi les professionnels de la santé à prendre des décisions éclairées.

À travers l'implémentation d'un système d'aide à la décision basé sur le raisonnement à partir de cas à l'aide de la plateforme JColibri, nous avons pu concrétiser cette approche dans un cadre pratique. Notre solution offre un moyen efficace d'intégrer l'intelligence artificielle dans la pratique médicale, en combinant l'expertise humaine avec les avantages du traitement automatisé des cas.

En résumé, notre mémoire a contribué à l'avancement des systèmes d'aide à la décision médicale en démontrant les avantages et les possibilités offerts par le raisonnement à partir de cas dans le diagnostic du diabète. Nous sommes convaincus que notre travail ouvre de nouvelles perspectives pour l'utilisation de l'intelligence artificielle dans le domaine médical, en fournissant des outils précieux pour les professionnels de la santé et en améliorant les résultats des patients.

En conclusion, notre mémoire constitue une contribution significative à l'amélioration des systèmes d'aide à la décision médicale et ouvre la voie à des avancées futures dans le domaine de l'intelligence artificielle en santé. Nous espérons que nos résultats inciteront d'autres chercheurs et praticiens à explorer les bénéfices du raisonnement à partir de cas dans le contexte de la prise de décision médicale, pour le bien-être des patients et l'amélioration des soins de santé.

# Bibliographie

- [1] MANSOUL ABDELHAK. “Aide à la décision médicale guidée par un processus d’extraction de connaissances”. Thèse de doct. Université de D’oran, 2018.
- [2] Dawit Hassen ABTEW. “Integrating Descriptive Modelling with Case Based Reasoning in Network Intrusion Detection .” Thèse de doct. the College of Natural Sciences of Addis Ababa University In Partial Fulfillment, 2015.
- [3] Jay E ARONSON, Ting-Peng LIANG et Richard V MACCARTHY. *Decision support systems and intelligent systems*. T. 4. Pearson Prentice-Hall Upper Saddle River, NJ, USA : 2005.
- [4] Aferhane ASMA. “Les systèmes d’aide à la décision médicale : Apprentissage automatique”. Thèse de doct. FACULTE DES SCIENCES ET TECHNIQUES FES – SAISS, 2018.
- [5] Fouad BENAMAR et Fethi BOUGUELMOUNA. “Raisonnement à Partir de Cas En Utilisant le Système jCOLIBRI”. In : ().
- [6] Lamine BENMIMOUNE. “Une approche pour la conception de systèmes d’aide à la décision médicale basés sur un raisonnement mixte à base de connaissance”. Thèse de doct. Université de Technologie de Belfort-Montbéliard, 2016.
- [7] Saroj Kr BISWAS et al. “An ANN based classification algorithm for swine flu diagnosis”. In : *Int. J. Knowl. Based Comput. Syst* 3.1 (2015), p. 31-42.
- [8] Henni Fouad BOURAHLA AFIF. *Le Raisonnement à Partir de Cas dans les Systèmes d’Aide à la Décision Médicale , Prise en Charge de l’Enfant Victime d’un Accident de la Route*. 2016/2017.
- [9] Eric BUIST. “Les éléments fondamentaux du raisonnement à base de cas”. In : *Université de Montréal* ().
- [11] Benjamin DEVÈZE et Matthieu FOUQUIN. “Case-Based Reasoning”. In : *Rapport des études en spécialité SCIA, école de l’ingénieur EPTA, France* (2004).
- [13] Krzysztof DYCZKOWSKI. *Intelligent medical decision support system based on imperfect information*. T. 735. Springer, 2018, p. 978-3.
- [14] Ziad EL BALAA et al. “Fm-ultranet : a decision support system using case-based reasoning, applied to ultrasonography”. In : *Workshop on CBR in the Health Sciences*. T. 37. Springer-Verlag ICCBR’03, NTNU, Trondheim, Norway. 2003, p. 0–3.
- [17] Lothar GIERL, Mathias BULL et Rainer SCHMIDT. “CBR in Medicine”. In : *Case-based reasoning technology : from foundations to applications* (1998), p. 273-297.

- [18] S GUESSOUM et al. “La modélisation informatique au service du raisonnement médical : le RàPC pour l’aide au diagnostic du cancer broncho-pulmonaire primitif (CBP)”. In : *Revue des Maladies Respiratoires* 34 (2017), A88.
- [20] Farhi MARIR et Ian WATSON. “Case-based reasoning : a categorized bibliography”. In : *The Knowledge Engineering Review* 9.4 (1994), p. 355-381.
- [21] Labed MARWA. “Classification pour diagnostic du diabète utilisant les algorithmes d’apprentissage.” Thèse de doct. UNIVERSITE BADJI MOKHTAR ANNABA, 2021.
- [23] Mr Hamza MEGUEHOUT. “Un raisonnement à partir de cas (RàPC) pour la traduction automatique du langage naturel”. Thèse de doct. UNIVERSITE BADJI MOKHTAR-ANNABA, 2019.
- [24] H MINTZBERG. “The structure of unstructured decision process”. In : *Administrative Science Quarterly* (1976).
- [25] M MORENO. “Développement des systèmes d’aide à la décision dans les cabinets de médecine générale en France”. Thèse de doct. univ-lorraine, 1915.
- [26] Nourhène BEN RABAH. “APPROCHE INTELLIGENTE À BASE DE RAISONNEMENT À PARTIR DE CAS POUR LE DIAGNOSTIC EN LIGNE DES SYSTÈMES AUTOMATISÉS DE PRODUCTION”. Thèse de doct. UNIVERSITE DE REIMS CHAMPAGNE, 2018.
- [27] Ivana RASOVSKA. “Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas : Application au diagnostic dans une plateforme d’e-maintenance.” Thèse de doct. Université de Franche-Comté, 2006.
- [28] RECIO-GARCÍA. “COLIBRI Studio Tutorial”. In : *Department of Software Engineering and Artificial Intelligence, University Complutense of Madrid, Madrid* (August 31,2008).
- [29] Juan A RECIO-GARCÍA, Belén DÍAZ-AGUDO et Pedro GONZÁLEZ-CALERO. “jcolibri2 tutorial”. In : *Department of Software Engineering and Artificial Intelligence, University Complutense of Madrid, Madrid* (2008).
- [30] Shaker H EL-SAPPAGH et Mohammed ELMOGY. “Case based reasoning : Case representation methodologies”. In : *International Journal of Advanced Computer Science and Applications* 6.11 (2015), p. 192-208.
- [31] H SIMON. “systems development in an emerging economy. Decision Support Systems”. In : (1977).
- [32] Abir SMITI et Zied ELOUEDI. “Overview of maintenance for case based reasoning systems”. In : *International Journal of Computer Applications* 975 (2011), p. 8887.
- [33] “Systèmes informatiques d’aide à la décision en médecine : panorama des approches utilisant les données et les connaissances”. In : (1914).
- [34] Daniel Kahsu TESFAY. “A CASE BASED REASONING SYSTEM FOR DIAGNOSIS OF MALNUTRITION FOR UNDER-FIVE YEAR CHILDREN : THE CASE OF TIRUNESHE BEJING”. Thèse de doct. St. Mary’s University In Partial Fulfillment of the Requirements for the Degree of Master of Science, 2022.

## Bibliographie

---

- [35] A TSOUKIAS. “decision theory to decision aiding methodology”. In : *DIMACS Technical report, Rutgers University* (2003).
- [38] Ian WATSON et Farhi MARIR. “Case-based reasoning : A review”. In : *The knowledge engineering review* 9.4 (1994), p. 327-354.
- [39] Wira Zanoramy ZAKARIA et Laila Mat KIAH. “Implementing a CBR Recommender for Honeypot Configuration using jCOLIBRI”. In : *Conference Paper. Conference : 3rd International Conference on Computer Science and Computational Mathematics (ICCSCM 2014)*. 2014.
- [40] Abdelhamid ZEMIRLINE. “Définition et fusion de systèmes diagnostic à l’aide d’un processus de fouille de données : Application aux systèmes diagnostics”. Thèse de doct. Télécom Bretagne, 2008.

# Webographie

- [10] *comprendre-le-diabete*. URL : <https://www.diabete.qc.ca/fr/comprendre-le-diabete/tout-sur-le-diabete/symptomes/les-symptomes/>.
- [12] *Diabetes prediction dataset*. fr. URL : <https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset>.
- [15] *Etapas du processus*. URL : [https://www.techno-science.net/glossaire-definition/Raisonnement-par-cas.html#ref\\_1](https://www.techno-science.net/glossaire-definition/Raisonnement-par-cas.html#ref_1).
- [16] *Ethique de la décision médicale*. 2014. URL : <https://www.cairn.info/revue-laennec-2014-4-page-40.htm>.
- [19] *Herbert Simons Decision Making Theory*. en. juin 2021. URL : <https://harappa.education/harappa-diaries/herbert-simons-decision-making-theory/>.
- [22] *Medical Decision Support Systems and Knowledge Sharing Standards*. URL : <http://what-when-how.com/medical%20informatics/medical-decision-support-systems-and-knowledge-sharing-standards/>.
- [36] *types-de-diabete*. URL : <https://appli.dinnosante.fr/fre/18/le-diabete/types-de-diabete>.
- [37] *types-de-diabete*. URL : <https://www.who.int/fr/news-room/fact-sheets/detail/diabetes>.