

---

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
CENTRE UNIVERSITAIRE BELHADJ BOUCHAIB D'AÏN-TÉMOUCHENT



Institut des Sciences  
Département de Mathématiques et de l'Informatique

## Mémoire

Pour l'obtention du Diplôme de Master en Informatique  
Option : Réseaux et Ingénierie des Données (RID)

Présenté par :

Mme.BELHAMIANI FATIMA ZOHRA

Mme.MAAMAR ZAHEIRA

---

# OUTIL INTELLIGENT D'AIDE À MAINTENANCE INDUSTRIELLE : RAISONNEMENT À PARTIR DE CAS ET ONTOLOGIE

---

Encadreur :

Melle. BOUHALOUAN DJAMILA  
Maitre Assistant "A" à C.U.B.B.A.T.

Soutenu en juin 2019

Devant le jury composé de :

---

Président : M. BOUCHAKOUR Hichem (MCB) C.U.B.B.A.T.

Examineurs : Mme SAIDI. S (M.A.A) C.U.B.B.A.T.

Encadrant : Melle. BOUHALOUAN Djamilia (M.A.A) C.U.B.B.A.T.

---

---

## *Remerciements*

*En préambule à ce mémoire nous remerciant **ALLAH El Karim** qui nous a aidé et nous a donné la patience et le courage pour finaliser et faciliter ce travail.*

*Nous souhaitons adresser nos remerciements les plus sincères à notre encadreur Mademoiselle **Bouhalouan Djamila** qui nous a encadré, tout le long de ce mémoire. Nous lui apportons aussi toute notre reconnaissance pour son aide consistante, ses conseils précieux et ses remarques objectives.*

*C'est grâce à elle que ce travail a pris cette forme.*

*Notre sincères remerciements aux membres du jury : Monsieur*

***BOUCHAKOUR.H** et Madame **SAIDI.S**, qui ont accepté de juger notre travail.*

*Sans oublier tous les enseignants du département Mathématique et Informatique, et les employés de l'université continue et centre universitaire d'Ain Temouchent.*

---

## *Dédicaces*

*Je tien à dédier ce travail :*

*Á l'être la plus chère à mon cœur "mère ", pour son soutien et son amour ;*

*Á la mémoire de mon défunt père ;*

*Á mon mari Baghdad pour sa contribution, son soutien, sa patience et encouragement, sans oublier mes enfants Wisseme et Mohammed el Amine ;*

*Á mes frères Miloud, Mourad, Mohammed et Hichem ;*

*Á mes sœurs Khadidja et Sakina ;*

*Á mes belles sœurs Souhila, Mokhtaria et Fatima ;*

*Et un grand merci à mon encadreur Melle Bouhalouan Djamila, aux membres du jury ,à tous les enseignants du département de Mathématiques et Informatique, et à tous mes collègues qui travaillent avec moi à l'université de formation continue.*

**BELHAMIANI Fatima Zohra**

---

## *Dédicaces*

*Je dédie ce travail aux personnes les plus chères à mon cœur ; A mes chers parents, qui n'ont jamais cessé de m'épauler, que Dieu les préserve et leur accorde une longue vie ;*

*A mes frères et mes sœurs ;*

*A mon mari Youcef pour son soutien, sa patience et encouragement ;*

*A ma fille Tesnime ;*

*A mon binôme qui m'a soutenu durant la réalisation de notre mémoire ;*

*A mes très chers amis et bien d'autres ;*

*A toute ma famille, à tous mes amis ;*

*Et un grand merci à mon encadreur, aux membres du jury et à tous les enseignants du département de Mathématiques et Informatique, qui m'ont accompagné durant tout mon parcours universitaire ;*

**MAAMAR Zaheira**

## Résumé

Raisonnement sur des cas existants ou réutiliser des expériences passées suite à l'apparition de nouveaux problèmes afin de les résoudre est très habituel. Vu que les problèmes quasi ou totalement similaires vont avoir des solutions quasi ou totalement similaires. Cette tendance actuelle qui est le raisonnement à partir de cas (RPC) qui est un outil puissant et intelligent de conception de systèmes, a été renforcé dans notre projet par l'intégration des ontologies. Nous nous focalisons dans le domaine de la maintenance industrielle et plus précisément les systèmes de refroidissement des véhicules. Chaque problème posé a des descripteurs appelés aussi symptômes et une ou des solutions, et à partir de là on crée une base de connaissance appelée base de cas qui a une structure ontologique.

Notre système consiste à apporter de l'aide aux utilisateurs ou décideurs pour la résolution des problèmes apparus en se basant sur ceux déjà stockés (cas précédents). L'intégration des ontologies afin de décrire le vocabulaire des cas et les indexer ainsi d'en déduire des nouveaux et enrichir la base de connaissance avait un impact remarquable et fertile. Nous nous sommes servis d'outils divers tels que l'outil protégé pour éditer les ontologies, NetBeans qui est un éditeur graphique supportant JAVA et l'environnement JENA qui a été intégré en JAVA qui permet de manipuler les documents RDF, RDFS, OWL et fournit un moteur d'inférence permettant le raisonnement sur les ontologies.

**Mots clés :** Raisonnement à base de cas, Ontologies, Maintenance industriel

## **Abstract**

Reasoning on existing cases or reusing past experiences following the appearance new problems in order to solve them is very usual. View that almost or completely similar problems will have solutions almost or totally Similar. This current trend which is reasoning from case (PRC) which is a powerful and intelligent system design tool, has been strengthened in our project by integrating ontologies. We focus in the field industrial maintenance and more precisely the cooling systems cars. Each problem has descriptors also called symptoms and one or more solutions, and from there we create a base of knowledge called base case that has an ontological structure.

Our system consists of providing help to users or decision-makers for the resolution of problems that have arisen based on those already stored (previous cases). The integration of ontologies to describe the vocabulary of cases and index them thus to deduce new ones and enrich the knowledge base had a remarkable and fertile impact. We used various tools such as the protected tool to edit ontologies, NetBeans which is a graphical editor supporting JAVA and the JENA environment which has been integrated into JAVA which allows to manipulate RDF, RDFS, OWL and provides an inference engine for ontology reasoning.

**Keywords:** Case-Based Reasoning, Ontologies, Industrial Maintenance

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Diagnostic et raisonnement à base de cas</b>	<b>4</b>
1.1 Introduction . . . . .	5
1.2 Le diagnostic . . . . .	5
1.2.1 Les modes de fonctionnement . . . . .	6
1.2.2 Le diagnostic industriel . . . . .	7
1.3 Les connaissances dans le diagnostic : . . . . .	8
1.3.1 La connaissance globale : . . . . .	8
1.3.2 La connaissance instantanée : . . . . .	8
1.4 Raisonnement à base de cas (RBC) : . . . . .	8
1.4.1 Approche de raisonnement à partir de cas : . . . . .	8
1.5 Représentation d'un cas : . . . . .	9
1.5.1 Structure du cas : . . . . .	10
1.5.2 Indexation du cas : . . . . .	10
1.6 Le carré d'analogie : . . . . .	11
1.7 Cycle du RBC : . . . . .	12
1.7.1 Phase "Elaboration" : . . . . .	13
1.7.2 Phase "Remémoration" : . . . . .	13
1.7.3 Phase "Adaptation" : . . . . .	15
1.7.4 Phase "Révision" : . . . . .	16
1.7.5 Phase "Mémorisation" : . . . . .	16
1.8 Outils et méthodes intégrant le raisonnement à base de cas : . . . . .	17
1.8.1 Datamining : . . . . .	17
1.8.2 Systèmes experts : . . . . .	17
1.8.3 Algorithmes génétiques: . . . . .	18
1.8.4 Réseaux de neurones : . . . . .	18

1.8.5	K plus proches voisins (k-ppv) : . . . . .	19
1.8.6	Ontologies : . . . . .	19
1.9	Conclusion : . . . . .	19
<b>2</b>	<b>Les ontologies</b>	<b>20</b>
2.1	Introduction: . . . . .	21
2.2	Notion d'ontologie : . . . . .	21
2.2.1	Quelques Définitions: . . . . .	21
2.2.2	Objectifs de développement d'une ontologie: . . . . .	22
2.2.3	Différence entre ontologie et base de connaissance: . . . . .	22
2.3	Composantes d'une ontologie : . . . . .	23
2.3.1	Concept (classe): . . . . .	23
2.3.2	Relation: . . . . .	23
2.3.3	Fonctions: . . . . .	23
2.3.4	Axiomes: . . . . .	23
2.3.5	Instances : . . . . .	23
2.4	Types d'ontologies: . . . . .	24
2.4.1	Typologie selon le niveau de conceptualisation : . . . . .	24
2.4.2	Typologie selon le niveau de formalisation utilisée : . . . . .	25
2.4.3	Typologie selon le niveau de détail de l'ontologie: . . . . .	25
2.4.4	Typologie selon le niveau de complétude : . . . . .	26
2.5	Cycle de vie d'une ontologie : . . . . .	27
2.6	Processus de construction d'une ontologie : . . . . .	28
2.6.1	Spécification (évaluation des besoins): . . . . .	29
2.6.2	Conceptualisation : . . . . .	29
2.6.3	Ontologisation: . . . . .	29
2.6.4	Opérationnalisation : . . . . .	29
2.7	Méthodes et méthodologies de construction d'ontologies: . . . . .	30
2.7.1	Stratégies de construction d'une ontologie: . . . . .	30
2.7.2	Méthodologies de construction : . . . . .	30
2.8	Langages d'ontologie : . . . . .	31
2.8.1	Langages traditionnels : . . . . .	31
2.8.2	Web standard ou basé sur xml : . . . . .	31
2.8.3	Langages d'interrogation des ontologies : . . . . .	34
2.8.4	Approches du raisonnement : . . . . .	34
2.9	Outils de développement d'ontologie . . . . .	36
2.9.1	PROTÉGÉ . . . . .	36
2.9.2	ODE (ONTOLOGY DESIGN ENVIRONMENT) . . . . .	36



2.9.3	OntoEdit . . . . .	36
2.9.4	WebOde . . . . .	37
2.9.5	DoE . . . . .	37
2.9.6	Framwork jena . . . . .	37
2.10	UML – OWL . . . . .	38
2.10.1	Le Langage UML . . . . .	38
2.10.2	Les règles de passage d'un modele UML vers OWL . . . . .	38
2.10.2.1	Le Concept de classe . . . . .	38
2.10.2.2	Le Concept de propriété . . . . .	39
2.11	Conclusion . . . . .	40
<b>3</b>	<b>Modélisation et implémentation du système Onto-master</b>	<b>41</b>
3.1	Introduction . . . . .	42
3.2	Partie I : Conception et Modélisation . . . . .	42
3.2.1	Domaine d'étude (description du système d'étude) . . . . .	42
3.2.2	Présentation du système de refroidissement . . . . .	42
3.2.3	Fonctionnement du système de refroidissement . . . . .	43
3.2.4	Eléments du système . . . . .	44
3.2.5	Rôle des éléments du système . . . . .	44
3.2.6	Architecture générale de notre application . . . . .	45
3.2.7	Connaissances . . . . .	45
3.2.8	Base de cas . . . . .	46
3.2.9	Ontologies . . . . .	47
3.2.10	Diagramme de cas d'utilisation . . . . .	48
3.3	Partie II : Mise en œuvre . . . . .	49
3.3.1	Environnement de programmation . . . . .	49
3.3.1.1	NetBeans 8.2 . . . . .	49
3.3.1.2	JENA . . . . .	50
3.3.1.3	JAVA . . . . .	51
3.3.1.4	Protégé 4.2 . . . . .	51
3.3.2	Étapes de création de notre ontologie "onto": . . . . .	52
3.3.3	Quelques Scénarios (guide d'utilisation) . . . . .	57
3.3.4	Exemple d'application . . . . .	64
3.4	Conclusion . . . . .	65
	<b>Conclusion générale</b>	<b>66</b>

# Table des figures

1.1	Transition d'un état de bon fonctionnement à un état de panne causée par une défaillance[ <a href="#">MOHA 09</a> ] . . . . .	7
1.2	L'ingénierie des connaissances à l'intersection des deux communautés des Sciences Cognitives et de l'Intelligence Artificielle [ <a href="#">DEND 14</a> ] . . .	9
1.3	Carré d'analogie[ <a href="#">BAND 10</a> ] . . . . .	11
1.4	Le cycle de raisonnement à partir de cas selon[ <a href="#">MILL 99</a> ] . . . . .	12
1.5	Types de l'adaptation dans les systèmes de RBC[ <a href="#">IVAN 06</a> ] . . . . .	16
2.1	Cycle de vie d'une ontologie[ <a href="#">BANE 07</a> ] . . . . .	28
2.2	Processus général de construction d'une ontologie [ <a href="#">BELA 15</a> ] . . . . .	28
2.3	Triplet RDF[ <a href="#">BENA 17</a> ] . . . . .	32
2.4	Les langages d'exploitation des ontologies[ <a href="#">GOME 99</a> ] . . . . .	33
2.5	Diagramme de Classe UML . . . . .	39
2.6	Graphe OWL correspondant à l'exemple de propriété OWL. [ <a href="#">MACA 02</a> ] . . . . .	39
3.1	Fonctionnement du systeme de refroidissement(a) . . . . .	43
3.2	Fonctionnement du systeme de refroidissement(b) . . . . .	43
3.3	Représentation générale de l'application . . . . .	45
3.4	Diagramme de classe UML de la base de cas . . . . .	46
3.5	Différents modèles conceptuels d'ontologie . . . . .	47
3.6	Diagramme de cas d'utilisation . . . . .	48
3.7	Ajout de la bibliothèque API jena à notre projet . . . . .	50
3.8	Création des classes . . . . .	52
3.9	Création des propriétés pour les classes . . . . .	52
3.10	Création des individus(instances) . . . . .	53
3.11	Attribution des valeurs aux instances . . . . .	53
3.12	Hierarchie des concepts de l'ontologie "onto" avec Protégé OWL . . . . .	54

## TABLE DES FIGURES

---

3.13	Interface OntoGraph . . . . .	55
3.14	Ontologie OWL . . . . .	56
3.15	Interface principale . . . . .	57
3.16	Authentification de l'opérateur . . . . .	58
3.17	Interface Accueil . . . . .	59
3.18	Résolution d'un problème . . . . .	60
3.19	Interface mise à jour . . . . .	62
3.20	Ajout d'un nouveau composant à la BC . . . . .	63
3.21	Ajout d'un nouveau symptôme à la BC . . . . .	63

# Liste des tableaux

2.1	Notion de classe en UML, graphe et code . . . . .	38
3.1	Cas similaire trouvé . . . . .	64

# Introduction Générale

## **1.Contexte du projet**

Le raisonnement à base de cas (RBC) occupe une place essentielle dans beaucoup de systèmes informatiques, C'est une approche de résolution des problèmes qui utilise des expériences passées, Chaque expérience appelée cas et se compose du problème et solution, et de là en construisant une base de cas d'un domaine spécifié ou appelé une base de connaissance.

La base de connaissance est une base de données de capacité de représentation de l'information plus élevée d'où les bases de données simples ne suffisent pas telles que les bases de données relationnelles, les bases de données déductives, actives . . . etc. Ainsi la requête d'interrogation de la base ne reprend pas au besoin du domaine de connaissance.

D'autre part la modélisation et la conception des connaissances d'un domaine se rend difficile avec les bases de données simples ce qui fait l'apparition de notion d'ontologie comme un modèle conceptuel partageable d'un domaine. L'ontologie d'un domaine permet d'intégrer facilement la représentation explicite de la base de connaissance.

Et donc, notre travail s'inscrit dans le cadre d'un projet de fin d'étude de master

2 en informatique option « Réseaux et Ingénierie des données » au sein du centre universitaire d'Ain Temouchent BELHADJ Bouchaib où nous avons conçu une application d'aide à la décision à partir de cas pour le diagnostic industriel.

### **2.Problématique et objectifs du projet**

Au cours de ce projet nous nous intéressons à la réalisation de la fonction maintenance en contexte industriel et plus particulièrement, à l'aide qui peut être apportée aux processus de résolution par la réutilisation des connaissances et les ontologies. Suite à l'avènement d'un nouveau problème, nous devons explorer la base de cas afin d'extraire les cas similaires pour la résolution de ce problème, dans le cas contraire proposer une nouvelle solution pour le résoudre.

Le but de ce projet est de réaliser un outil intelligent d'aide à maintenance industrielle en utilisant le raisonnement à partir de cas intégrant les ontologies.

### **3.Approche utilisée – Contribution –**

Dans notre approche de raisonnement à partir de cas intégrant les ontologies, nous nous sommes basées sur une base de cas exemple d'un système de refroidissement d'un moteur de véhicule (voir le détail dans le chapitre 3).

La base de cas servira à collecter les informations du système de refroidissement du moteur tel que le composant, le problème posé, les solutions, les causes ainsi que les méthodes de résolution afin d'effectuer une recherche de cas similaire au problème à résoudre et extraire la solution ou adapter et stocker le nouveau cas.

L'ontologie servira d'abord à la formalisation et la modélisation du domaine de système de refroidissement. En effet, nous avons choisi le langage OWL pour la représentation des modèles.

Le langage SPARQL est utilisé pour extraire la solution du problème posé avec une interrogation de la base de cas.

### **4.Description du contenu du mémoire**

La suite de ce document est organisée comme suit :

- Tout d’abord, au niveau du chapitre 1 nous allons présenter le raisonnement à partir de cas, le mode de fonctionnement, les diagnostics industriels, le cycle de raisonnement à partir de cas et enfin les outils intégrant le raisonnement à partir de cas.
- Le chapitre 2 présente les ontologies principalement les définitions, les méthodes de construction ainsi que les langages des ontologies.
- Dans le chapitre 3 nous détaillons la conception puis la réalisation de l’outil que nous avons développé.
- Finalement, nous terminons par une conclusion au niveau de laquelle nous citons les difficultés rencontrées lors de la réalisation de ce projet ainsi que quelques perspectives qui pourront compléter ce qui a été effectivement réalisé.

Chapitre **1**

# Diagnostic et raisonnement à base de cas

## Sommaire

---

1.1	Introduction . . . . .	5
1.2	Le diagnostic . . . . .	5
1.3	Les connaissances dans le diagnostic : . . . . .	8
1.4	Raisonnement à base de cas (RBC) : . . . . .	8
1.5	Représentation d'un cas : . . . . .	9
1.6	Le carré d'analogie : . . . . .	11
1.7	Cycle du RBC : . . . . .	12
1.8	Outils et méthodes intégrant le raisonnement à base de cas : . . . . .	17
1.9	Conclusion : . . . . .	19

---



## 1.1 Introduction

Le Raisonnement à base de cas (RCB) est une méthode d'apprentissage et de résolution de problèmes basée sur les expériences passées.

Un système informatique de RCB met en œuvre une base de cas composée d'expériences passées dans laquelle peuvent être recherchées des expériences similaires au problème à résoudre.

Dans ce chapitre nous nous présentons le diagnostic de pannes sur les équipements industriels, nous donnons les notions de base concernant le diagnostic ainsi que la relation entre diagnostic et raisonnement à base de cas, cette technique qui est largement utilisée dans le domaine du diagnostic.

## 1.2 Le diagnostic

Avant de définir le diagnostic, nous introduisons quelques notions de bases qui l'exploitent Selon l'AFNOR (Association Française de NORmalisation) dans [AFNO 01]

- **Un bien** : est tout élément, composant, mécanisme, sous-système, unité fonctionnelle, équipement ou système qui peut être considéré individuellement. Ces biens représentent des équipements industriels qui sont constitués de plusieurs composants susceptibles d'être défectueux voir tomber en panne.
- **Un composant** : est un élément discret d'un équipement. C'est l'unité de description minimale de l'analyse systémique. C'est l'élément de base qui est interconnecté avec d'autres composants de même nature ou de nature différente dans un espace mitoyen ou lointain. Le domaine du diagnostic fait référence aux notions d'observation, de panne, de défaillance et de symptôme[PIEC 03].
- **Une observation** : est une information obtenue sur un équipement [AFNO 01]. Grâce à l'observation, on peut déduire l'état des composants. Les observations interviennent lors de l'apparition des symptômes.
- **Un symptôme** : est un phénomène qui survient sur un dispositif et qui révèle un dysfonctionnement [AFNO 01].

Il est également fréquent de regrouper les symptômes en fonction du dysfonctionnement auxquels ils sont liés : on parle alors de syndrome, L'apparition de symptômes cause une défaillance.

### 1.2.1 Les modes de fonctionnement

On distingue les modes de fonctionnement :

- Normale
- Dégradé
- Et défaillant
- **Une défaillance** : est définie comme la cessation de l'aptitude d'un bien à accomplir une fonction requise [AFNO 01]. C'est un évènement correspondant à un dysfonctionnement du service lié à un fonctionnement anormal ou plus exactement non conforme aux spécifications à un instant donné
- **Une dégradation** : est une évolution irréversible d'une ou plusieurs caractéristiques d'un bien liée au temps, à la durée d'utilisation ou à une autre cause externe [AFNO 01].

De ce fait, la défaillance diffère de la dégradation, qui est une évolution de l'état. Le fonctionnement de l'équipement est amené à évoluer dans le temps. Cette évolution peut se traduire par l'éloignement du mode de fonctionnement, qui est initialement en mode de fonctionnement normal (fonctionnement en mode nominal), à un mode de fonctionnement dégradé. En effet, une dégradation peut aller jusqu'à une panne.

- **Une panne** : est définie par un état d'un bien inapte à accomplir une fonction requise, excluant l'inaptitude due à la maintenance préventive ou à d'autres actions programmées ou à un manque de ressources extérieures [AFNO 01]. Elle peut également désigner une anomalie, condition anormale diminuant ou supprimant l'aptitude d'une entité fonctionnelle à réaliser une activité requise. [MOHA 09]

La figure suivante montre le passage d'un état de bon fonctionnement à un état de panne en passant par la dégradation et la défaillance.

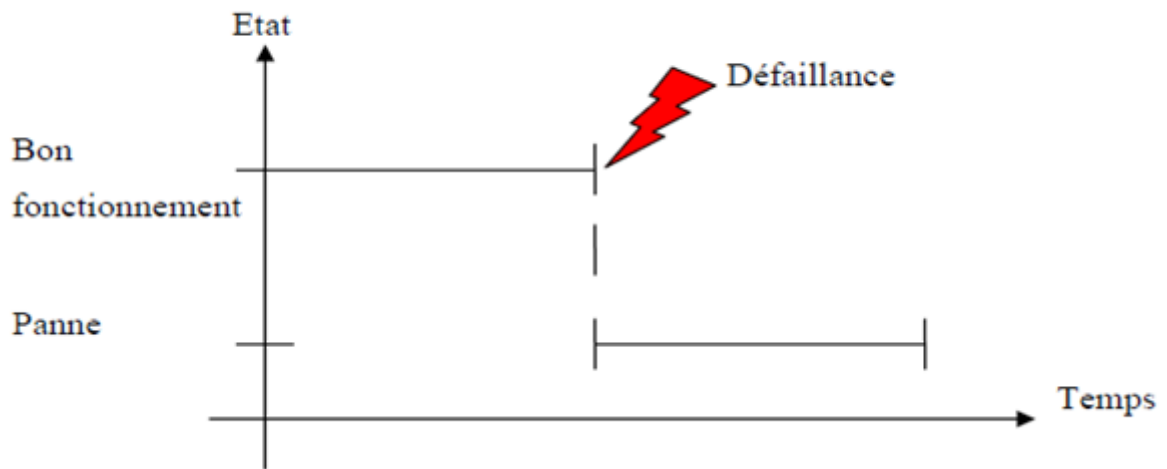


FIGURE 1.1 – Transition d'un état de bon fonctionnement à un état de panne causée par une défaillance[MOHA 09]

### 1.2.2 Le diagnostic industriel

C'est l'identification de la cause probable de la (ou des) défaillance(s) à l'aide d'un raisonnement logique fondé sur un ensemble d'informations provenant d'une inspection, d'un contrôle ou d'un test. Il y a deux tâches essentielles en diagnostic :

- L'observation des symptômes de défaillance.
- L'identification de la cause de la défaillance à l'aide d'un raisonnement logique fondé sur l'observation [BAND 10].

Le diagnostic est l'étape qui consiste à rechercher quelle est l'origine du défaut. Cette recherche n'a lieu qu'à la condition qu'un défaut ait été détecté [BAND 10]. Une définition normalisée du diagnostic que nous prenons comme référence a été proposée par l'AFNOR [AFNO 01] : « ce sont les actions menées pour la détection de la panne, sa localisation et l'identification de la cause ».

Selon la même source, la localisation de la panne représente les actions menées en vue d'identifier où se situe le fait générateur de la panne (la cause).[BAND 10].

## 1.3 Les connaissances dans le diagnostic :

Dubuisson et al [DUBU 01] considèrent deux types de données de diagnostic : les données numériques (exploitation des observations issues des capteurs par exemple) et les données symboliques (connaissances sur le système étudié). [BAND 10].

Il ya deux types de connaissances :

### 1.3.1 La connaissance globale :

Elle peut être qualifiée de connaissance à priori sur le système, cette connaissance s'appuie sur des expériences passées du système.

### 1.3.2 La connaissance instantanée :

Elle concerne l'ensemble des biens impliqués à un instant donné afin de prendre une décision et l'exploiter ; cette connaissance est issue des observations qui peuvent être numériques ou symboliques. [MOHA 09]

## 1.4 Raisonnement à base de cas (RBC) :

### Définition :

Le raisonnement à base de cas « RBC » a émergé depuis les années 1980. C'est une approche de l'intelligence et de psychologie orientée vers la résolution de problèmes basée sur les expériences passées. [BAND 11]

Un nouveau problème est résolu en cherchant un cas passé le plus similaire et en l'utilisant pour la résolution de la nouvelle situation problématique.[NOUR 17]

### 1.4.1 Approche de raisonnement à partir de cas :

Le RBC touche principalement deux types de communautés :

Celles de l'intelligence artificielle et des sciences cognitives[BAND 10]

L'IC (L'ingénierie des connaissances) se place à l'intersection de deux communautés de recherche[MOHA 09]

#### – L'intelligence artificielle (IA) :

Est un domaine de recherche permettant d'élaborer des systèmes intelligents.

L'IA est la « recherche de moyens susceptibles de doter les systèmes informatiques de capacités intellectuelles comparables à celles des êtres humains »

[MOHA 09].

– **Les sciences cognitives :**

Elles étudient la théorie de la compréhension, la résolution et l'apprentissage des problèmes par les hommes et essaient donc de modéliser les connaissances, les expériences et le raisonnement d'un être humain afin d'expliquer les motifs et le fonctionnement de son cerveau [BAND 10].

– **L'ingénierie des connaissances :**

C'est le domaine qui correspond à l'étude des concepts, méthodes et techniques permettant de modéliser et/ou d'acquérir les connaissances pour des systèmes réalisant ou aidant les humains à réaliser des tâches se formalisant à priori peu ou pas [MOHA 09].

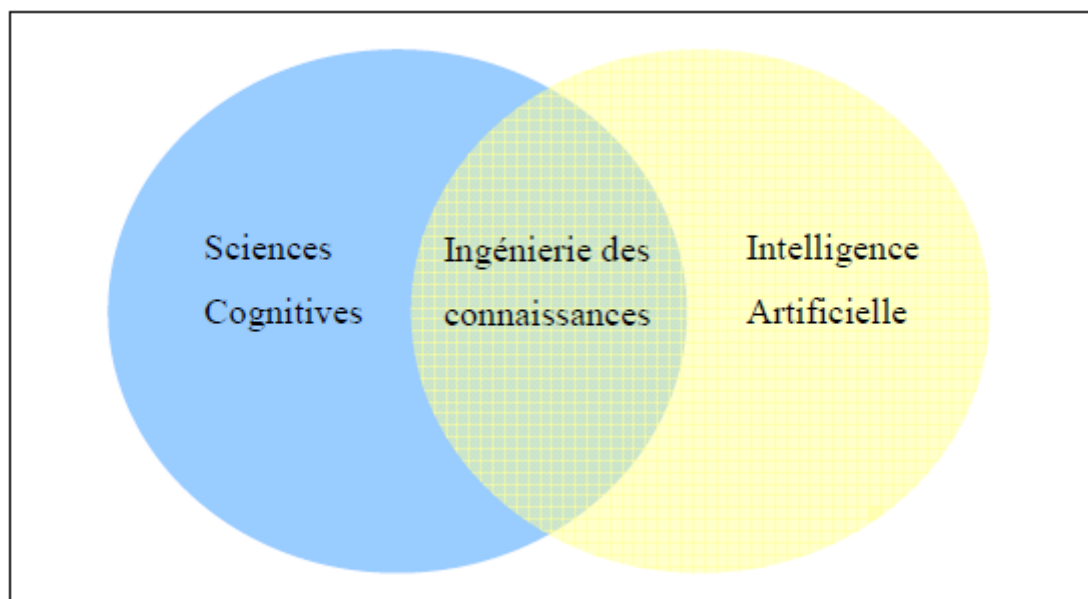


FIGURE 1.2 – L'ingénierie des connaissances à l'intersection des deux communautés des Sciences Cognitives et de l'Intelligence Artificielle [DEND 14]

## 1.5 Représentation d'un cas :

Un cas est une expérience représentée par une connaissance. Cette expérience constitue une leçon permettant au système de RBC de résoudre des problèmes de différentes natures. [DEND 14]

Un cas dans une base de cas représente différents types de connaissances qui peuvent être stockées selon différents formats de représentations.

### 1.5.1 Structure du cas :

Un cas est généralement composé de deux espaces disjoints : l'espace des problèmes et l'espace des solutions.[BAND 11]

- L'espace problème concerne la partie dans laquelle on trouve les objectifs à atteindre « pb ».
- L'espace solution regroupe la description de la solution apportée par le raisonnement, sa justification, son évaluation, ainsi que les étapes qui ont menés à cette solution « sol ( pb ) ». Cas = ( pb , sol ( pb ) ).

On peut distinguer deux types de cas : cas source et cas cible.

1. Un cas source est représenté par un couple ( srce , Sol ( srce ) ).
2. Un cas cible est représenté par le couple ( cible , Sol ( cible ) ), où Sol ( cible ) est inconnue et pour laquelle on voudrait lui apporter un résultat.[BAND 11]  
Selon les applications de RBC, son problème et sa solution, sont décrits par un ensemble de descripteurs.[BAND 10]
3. Un descripteur est généralement caractérisé par une paire  $d = (a, v)$ . Où :  
a : est un attribut défini par un nom.  
v : est la valeur qui lui est associée.

Comme les cas sont représentés par un ensemble de descripteurs alors :

- dsi (pour  $i = 1, \dots, n$ ) : représente les descripteurs de la partie problème du cas source « srce » ;
- dcj (pour  $i = 1, \dots, n$ ) : représente les descripteurs de la partie problème du cas cible « cible » ;
- Dsi (pour  $i = 1, \dots, m$ ) : représente les descripteurs de la partie solution du cas source « Sol(srce) » ;
- Dcj (pour  $i = 1, \dots, m$ ) : représente les descripteurs de la partie solution du cas cible «Sol(cible) ».[DEND 14]

### 1.5.2 Indexation du cas :

Les cas sont organisés dans une mémoire appelée base de cas. Afin de faciliter cette organisation et ainsi la recherche du cas le plus approprié au problème posé, il faut désormais les indexer. Il est à noter que lors de la recherche des cas, c'est la partie problème qui va être sollicitée. Or, cette partie problème est décrite par un ensemble de caractéristiques pertinentes nommées « indices ». Ces indices vont déterminer dans quels contextes et dans quelles situations les cas vont être recherchés et retrouvés pour les proposer au problème rencontré.

Alors il faut trouver le moyen de bien manipuler ces indices pour une configuration optimale. Pour cela, il y a plusieurs méthodes d'indexation : manuelles ou automatiques.[DEND 14]

## 1.6 Le carré d'analogie :

En 1996 Mille et ses co-auteurs [STAN 09], ont introduit la présentation du RBC en utilisant le carré d'analogie. Ce carré d'analogie exprime[DEND 14] :

- D'une part, les liens entre la description et la solution du cas source de la base de cas et du cas cible représentant un nouveau problème à résoudre (similarité entre deux problèmes). Dans ce cas là, la solution du cas cible est adaptée en fonction de la similarité et les descripteurs des cas sources similaires de la base de cas et sont adaptés au cas cible.
- D'autre part, le lien entre la description d'un cas et sa solution (la trace du raisonnement menant à la solution).[MOHA 09]

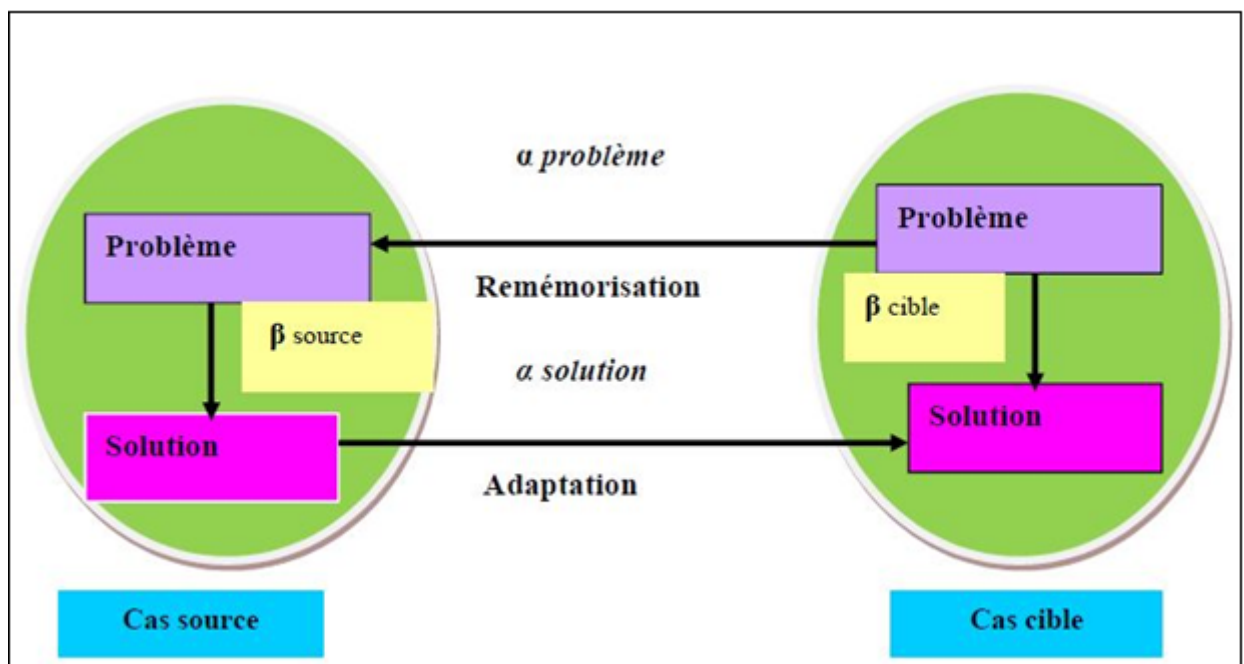


FIGURE 1.3 – Carré d'analogie[BAND 10]

$\alpha$  : similarité entre deux problèmes (cas)

$\beta$  : trace du raisonnement menant à la solution

## 1.7 Cycle du RBC :

Le RBC dispose d'un cycle dont le nombre de phases varie selon les différentes sources bibliographiques. Il peut être composé de trois, quatre ou cinq phases.

Fuchs et al[FUCH 99] déterminent trois phases à savoir la remémoration, l'adaptation et la mémorisation.

Les premiers auteurs à avoir décrit le cycle du RBC sont Aamodt et Plaza [AAMO 94] et le composent de quatre phases : la remémoration (ou recherche du cas similaire), l'adaptation (ou la réutilisation du cas retrouvé), la validation (ou la révision du cas sélectionné) et la mémorisation (ou l'apprentissage). Quant à Mille[MILL 06], il ajoute une phase préliminaire d'élaboration au début du cycle. La Figure 2 montre le cycle de RBC avec ces cinq phases.[MOHA 09]

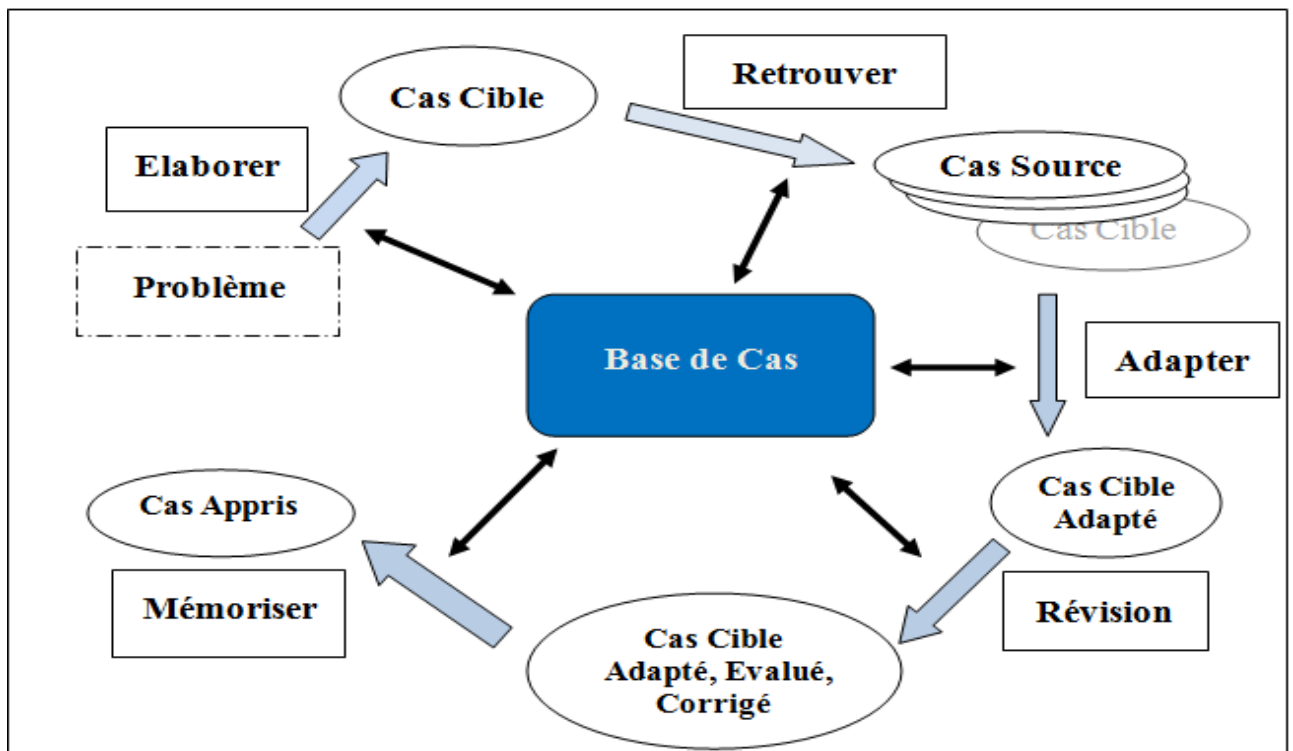


FIGURE 1.4 – Le cycle de raisonnement à partir de cas selon[MILL 99]

**Processus du RBC :** il s'effectue selon les étapes suivantes :

- **La phase d'élaboration :** dans laquelle le cas cible est construit en complétant ou filtrant la description d'un problème à partir d'une description éventuellement incomplète.
- **La phase de remémoration :** des cas sources à partir de la base de cas en recherchant des correspondances entre descripteurs des cas sources et du cas à résoudre (cible) .



- **La phase d'adaptation** : consiste à construire une solution au problème du cas cible inspirée de la solution du (des) cas source(s) le(s) plus similaire(s) .
- **La phase de révision** : de la solution proposée en cas d'une éventuelle solution insatisfaisante, alors il serait possible de la corriger. Dans ce cas, la solution est évaluée dans le monde réel en s'appuyant soit sur l'utilisateur, un expert humain, les connaissances du domaine ou sur un processus automatique
- **La phase de mémorisation** : consiste à stocker un nouveau cas résolu dans la base de cas si ce stockage est jugé opportun afin d'enrichir la mémoire du système.

Détaillons maintenant les phases su-citées :

### 1.7.1 Phase " Elaboration " :

La phase d'élaboration consiste à partir de l'entrée du système du RBC, à construire le problème cible : Élaboration : pré-cible  $\rightarrow$  cible

Où, la pré-cible dénote l'entrée du système de RCB, c'est-à-dire, le problème cible avant sa mise en forme. La mesure de similarité réalisée dans la phase de remémoration exprime ainsi l'importance et l'influence des descripteurs de problèmes pour sélectionner un cas dont la solution sera adaptable en vue de la résolution d'un problème cible. Cette phase nécessite deux tâches principales, création d'un cas et préparation de cas.

Lors de l'élaboration, un nouveau cas créé et la description du problème est ensuite complétée en collectant d'autres informations pertinentes pour le problème à résoudre. La deuxième tâche s'agit de préparer le cas, c'est-à-dire choisir les indices adéquats pour la recherche des cas. Ces indices sont les descripteurs du cas considérés comme pertinents pour la résolution du problème et utilisés comme critères de recherche dans la base de cas.

Ceci est en vue de la remémoration en choisissant les indices à utiliser pour se remémorer un cas passé. Le rôle de l'élaboration est donc de construire les descripteurs influents afin que la recherche des cas proche puis l'adaptation soit efficace.[[BAND 10](#)]

### 1.7.2 Phase "Remémoration" :

La phase de remémoration est une étape importante dans le système de RBC. L'une des hypothèses majeures dans le RBC est de dire que des expériences similaires peuvent guider de futurs raisonnements, des résolutions de problèmes et permettre un apprentissage de la base de cas. La remémoration est donc le processus qui permet

de retrouver des cas sources similaires au cas cible. Les cas sources similaires sont des cas qui ont besoin de moins d'adaptation. La mesure de similarité cherche la ressemblance entre les descripteurs des sources et cibles. Les tâches de remémoration consistent à « rechercher » des cas similaires, à les « apparier », c'est-à-dire les « évaluer et calculer la similarité », et enfin à « sélectionner » le cas source le plus adapté.

**Techniques de recherche des cas similaires :** Il existe différentes mesures pour calculer la similarité, ces mesures peuvent être locales ou globales. [MOHA 09]

**a- Similarités locales** [BAND 10] :

Elles sont basées sur les caractéristiques du cas. Elles dépendent du type des caractéristiques et des rangs des valeurs des caractéristiques. Généralement, le calcul des similarités locales dépend du type de descripteur et est basé sur la distance.

- Pour les valeurs de descripteurs numériques :

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

- Pour les valeurs de descripteurs symboliques (mono-valeurs) :

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

- Pour les valeurs de descripteurs symboliques (multi-valeurs)

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

- Pour les valeurs de descripteurs taxonomiques

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

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.

**b- Similarités globales :** elles sont calculées au niveau des cas ou des objets en agrégeant les similarités locales. Plusieurs similarités globales sont utilisées dans les systèmes de RBC et aucune d'elles n'est universelle et restent dépendantes du domaine concerné.

On peut citer :

- **Weighted Block-City :**

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

- **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.

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

- **Maximum based** :

$$sim(A, B) = max_i w_i sim(a_i, b_i)$$

Où :

$n$  : est le nombre d'attributs,

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

$Sim_i$  : est la similarité locale calculée pour l'attribut  $i$ .

### 1.7.3 Phase "Adaptation" :

La phase d'adaptation dans le cycle du RBC est le processus proposant une solution à un nouveau problème à partir des solutions appartenant aux cas sources mémorisés [LOPE 05].

En 1999 Fuchs et al. . . considèrent l'adaptation comme un plan dont l'état initial est la solution de départ et l'état final est la solution adaptée.[FUCH 99]

Lieber et al. . . [LIEB 04] considèrent que la phase d'adaptation consiste à effectuer un raisonnement par analogie.L'adaptation termine « l'inférence analogique» en calculant la solution possible au problème du cas cible inspirée de la solution du cas source le plus similaire.

Cette phase peut se faire soit via une intervention humaine (manuelle) soit d'une manière automatique à l'aide d'algorithmes, de méthodes, de formules, de règles, ... Concernant l'adaptation automatique, Wilke et Bergmann[WILK 98] déterminent les principaux types d'adaptation automatique :

- **L'adaptation générative** : part du fait que nous disposons de toutes les connaissances pour résoudre le problème à partir de zéro. Le cas retrouvé retrace le raisonnement ayant mené à la solution ;
- **L'adaptation transformationnelle** : est contraire à la précédente, c'est-à-dire qu'on ne dispose pas de toutes les connaissances pour résoudre le problème à partir de zéro .
- **L'adaptation compositionnelle** : utilise deux ou plusieurs cas similaires mémorisés pour effectuer l'adaptation en composant les différentes solutions proposées.
- **L'adaptation hiérarchique** : où les cas sont organisés à plusieurs niveaux dans la hiérarchie de généralisation.

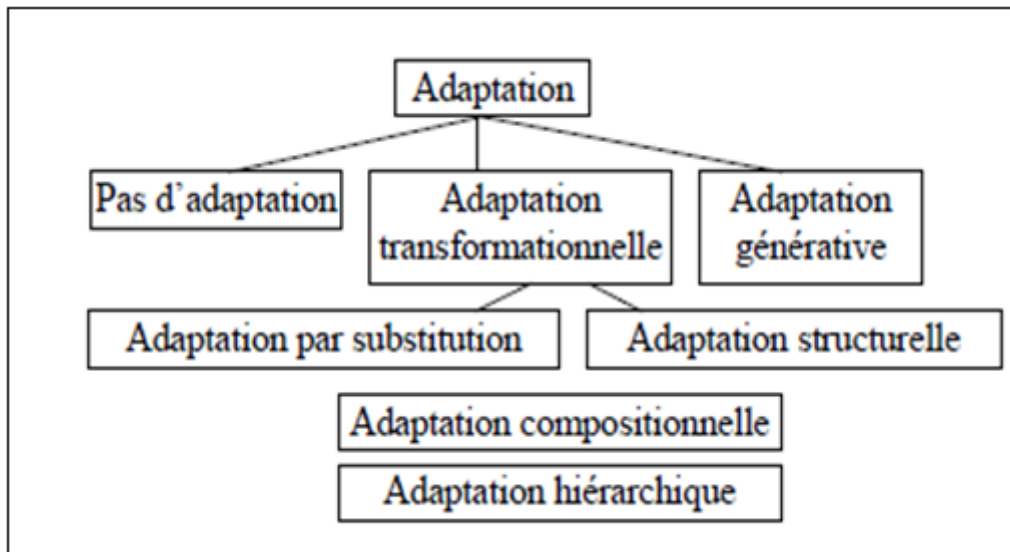


FIGURE 1.5 – Types de l'adaptation dans les systèmes de RBC[IVAN 06]

#### 1.7.4 Phase "Révision" :

Au cours de la phase de révision, la solution proposée à l'issue de la phase d'adaptation sera évaluée. Cette évaluation concerne plusieurs actions pouvant être employées [MILL 99] :

- Tester la solution proposée dans le monde réel .
- Faire une introspection dans la base de cas en utilisant l'ensemble des descripteurs de problème et de solution afin de vérifier que les cas similaires ont donné entière satisfaction.
- Utiliser une autre méthode d'évaluation de la solution (simulateur, système expert classique,...)[DEND 14]

Dans tous les cas, s'il y a divergence entre ce qu'a été construit comme solution et ce qui aurait fallu faire, il est important de :

- Corriger de façon à ce que le cas puisse être mémorisé comme un cas associé à une solution correcte.
- Tracer éventuellement, l'écart et de l'expliquer sous forme d'exception (ce qui revient à particulariser une forme d'adaptation en zone plus petite pour la classe de solution)[BAND 10].

#### 1.7.5 Phase "Mémorisation" :

Lorsqu'un cas a été validé par l'utilisateur, il est important de le sauvegarder dans la base de cas pour une réutilisation ultérieure. La mémorisation est la phase de stockage et d'enrichissement de la base de cas. Il est intéressant de pouvoir

sélectionner les cas à mémoriser. En effet, certains cas peuvent être très proches, et donc la mémorisation peut conduire à une duplication des informations de la base.

## 1.8 Outils et méthodes intégrant le raisonnement à base de cas :

Nous n'allons pas donner une liste exhaustive des techniques, mais on peut citer :

- Datamining ;
- Algorithmes génétiques ;
- Systèmes à base règle ;
- Systèmes experts ;
- Réseaux de neurone ;
- Ontologies ;
- ..... et d'autre.

### 1.8.1 Datamining :

Le Datamining est l'extraction des informations utiles, et la découverte des connaissances et des modèles cachés dans des bases de données hétérogènes. C'est un processus itératif et non-trivial de découverte d'informations implicites, précédemment inconnues et potentiellement utiles. Le Datamining s'adresse à la phase d'extraction et découverte de connaissances à partir de données, en utilisant le savoir-faire de plusieurs domaines : l'intelligence artificielle, l'apprentissage, la reconnaissance de modèles, l'acquisition de connaissances pour les systèmes experts, la visualisation des données ... etc.[[MEKR 09](#)]

### 1.8.2 Systèmes experts :

On peut définir le système expert comme étant un système informatique qui imite la démarche de la personne compétente dans un domaine donné, quelle que soit la méthode de raisonnement qu'elle utilise. De plus, il doit être interactif, capable de dialoguer avec ses utilisateurs et d'expliquer ses raisonnements[[CLER 01](#)].

Un système expert se compose de :

**a- Base de connaissances :**

Elle est élaborée à partir de l'expertise d'un spécialiste, elle se compose d'une base de faits et d'une base de règles[NEGR 91].

– **Base de règles :**

Elle contient les connaissances expertes (règles de l'expert) qui sont représentées généralement par des règles de production s'écrivant sous la forme : " Si Condition Alors Action"

– **Base de faits :**

C'est l'ensemble des propositions connues du système à un moment donné. C'est la mémoire de travail du système expert. Son contenu dépend du problème traité. La base de faits intègre deux types de faits : les faits permanents du domaine et les faits déduits par le moteur d'inférences qui sont propres au cas traité[NEGR 91].

**b- Le moteur d'inférences :**

Est un programme qui utilise les règles définies dans la base de connaissances pour résoudre un problème particulier décrit par des faits.

**c- L'interface**

L'interface pour l'aide à l'acquisition des connaissances fournies par l'expert peut être plus ou moins sophistiquée, l'accent étant souvent mis sur une syntaxe des règles la plus proche possible du langage naturel[NEGR 91].

### 1.8.3 Algorithmes génétiques :

Les algorithmes génétiques sont des techniques d'optimisation de type méta-heuristique fondées sur les principes de l'évolution biologique. C'est le type le plus robuste des algorithmes évolutionnaires [STAN 09].

L'inconvénient majeur de l'algorithme génétique est son temps d'exécution qui est assez long. Il n'y a pas de garantie quant à l'obtention de la solution optimale au problème posé en un temps fini et sa complexité augmente en fonction du nombre de règles utilisées.[BENB 12]

### 1.8.4 Réseaux de neurones :

Les Réseaux de neurones sont des modèles prédictifs non-linéaires. Ils sont à l'origine d'une tentative de modélisation mathématique du cerveau humain [BELLA 99].

Ils utilisent des données existantes avec un résultat connu pour former un modèle qui peut ensuite être utilisé pour faire des prédictions sur des données avec des résultats inconnus. Mais parfois les résultats ne sont pas explicites, ce qui est

intolérable pour certaines applications comme le diagnostic médical.[BENB 12]

### 1.8.5 K plus proches voisins (k-ppv) :

Les k plus proches voisins sont une méthode de raisonnement à partir de cas, dédiée à la classification. Les k-ppv permettent de prendre des décisions en recherchant des cas similaires déjà résolus[HAST 01].

### 1.8.6 Ontologies :

L'ontologie constitue un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné et de là on développe un système de raisonnement à base de cas intégrant ces ontologies.

Cette approche va être largement détaillée dans le prochain chapitre où nous allons présenter tous les concepts et principes la concernant.

## 1.9 Conclusion :

Pour développer un système de diagnostic pour un équipement industriel, notre choix s'est porté sur le raisonnement à base de cas (RCB), Car il permet de raisonner sur un nombre restreint des cas , et à travers ce chapitre nous avons présenté les notions de base liées au diagnostic tel que les pannes, les défaillances, ... etc, ainsi nous avons présenté les connaissances dans le diagnostic et leurs types tels que la connaissance globale et instantanée.

Puis, nous avons décrit d'une manière détaillée l'approche de (RCB). En effet, cette approche touche deux types de communauté telles que l'intelligence artificielle et les sciences cognitives et à l'intersection, l'ingénierie des connaissances.

Et enfin nous avons cité quelque outils et méthodes intégrant le raisonnement à base de cas et parmi lesquels, les ontologies qui seront le sujet de chapitre2 où nous allons cerner tous leurs principes.

# Chapitre 2

## Les ontologies

### Sommaire

---

2.1	Introduction : . . . . .	21
2.2	Notion d'ontologie : . . . . .	21
2.3	Composantes d'une ontologie : . . . . .	23
2.4	Types d'ontologies : . . . . .	24
2.5	Cycle de vie d'une ontologie : . . . . .	27
2.6	Processus de construction d'une ontologie : . . . . .	28
2.7	Méthodes et méthodologies de construction d'ontologies : . . . . .	30
2.8	Langages d'ontologie : . . . . .	31
2.9	Outils de développement d'ontologie . . . . .	36
2.10	UML – OWL . . . . .	38
2.11	Conclusion . . . . .	40

---



## 2.1 Introduction :

Les ontologies sont souvent capturées dans des langages de représentation de la connaissance (KR : Knowledge Representation) qui proviennent des théories de l'intelligence artificielle, les ontologies sont des architectures de concepts, non des listes organisées de termes.

Dans ce chapitre, Nous donnons premièrement des notions de base sur les ontologies, les différentes composantes, les méthodologies de construction d'ontologie et les différents modes de raisonnement ainsi que des outils de développement d'ontologie. Finalement, nous entamons le passage d'un modèle UML vers OWL.

## 2.2 Notion d'ontologie :

Ce terme est construit à partir des racines grecques «ontos» qui veut dire ce qui existe, l'être, et « logos » qui veut dire l'étude, d'où « l'étude de L'être ». [LORA 08]

### 2.2.1 Quelques Définitions :

Une ontologie, en philosophie, est une théorie à propos de la nature de l'existant, des types de choses qui existent et des types de leurs liens. Dans le Web sémantique, une ontologie est vue comme un ensemble de connaissances, y compris le vocabulaire et les relations sémantiques, avec quelques règles simples d'inférence et de logiques relatives à des sujets particuliers. [HACI 14]

- Dans le cadre de l'intelligence artificielle, Neches et ses collègues, furent les premiers à proposer une définition à savoir :  
« Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire ». [NECH 91]
- D'après Studer, l'ontologie est une « Spécification formelle et explicite d'une conceptualisation partagée ». [STUD 98]
  - Formelle : lisible par une machine.
  - Explicite : la définition explicite des concepts utilisés.
  - Conceptualisation : le modèle abstrait d'un phénomène du monde réel.
  - Partagée : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs. [MACA 02]
- Pour Guarino et Garetta, « une ontologie est une spécification rendant partiellement compte d'une conceptualisation ». [GUAR 95]

- En 1997, Swartout et ses collègues ont proposé la définition suivante :« une ontologie est un ensemble de termes structurés de façon hiérarchique, conçue afin de décrire un domaine et qui peut servir de charpente à une base de connaissances ». [SWAR 97]
- Cette notion est développée par Gomez comme :« une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances ». [GOME 99]

### 2.2.2 Objectifs de développement d'une ontologie :

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné.

Les ontologies sont employées dans l'intelligence artificielle, le Web sémantique, le génie logiciel, l'informatique biomédicale et l'architecture de l'information pour une représentation de la connaissance d'un domaine . Donc une ontologie est développée Pour les raisons suivantes :

- Partager la compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels.
- Permettre la réutilisation du savoir sur un domaine
- Expliciter ce qui est considéré comme implicite sur un domaine
- Distinguer le savoir sur un domaine du savoir opérationnel
- Analyser le savoir sur un domaine. [BELA 15]

### 2.2.3 Différence entre ontologie et base de connaissance :

Une différenciation claire entre « ontologie » et « base de connaissances » devait se faire à partir de son rôle. Une ontologie fournit un système de concepts qui sont utilisés pour construire une base de connaissances par-dessus ; par conséquent, une ontologie peut être une spécification de la conceptualisation du monde cible que se fait l'ingénieur qui construit la base de connaissances, donc un méta système d'une base de connaissances traditionnelle. [HACI 14 ]

## 2.3 Composantes d'une ontologie :

Partant du principe que les ontologies fournissent un vocabulaire commun d'un domaine et définissent la signification des termes et des relations entre elles. La connaissance dans les ontologies est principalement formalisée en utilisant les cinq types de composants selon Gomez à savoir : **concepts** (ou classes), **relations** (ou propriétés), **fonctions**, **axiomes** (ou règles) et **instances** (ou individus). [GOME 99]

### 2.3.1 Concept (classe) :

Représente les objets ou les notions d'où la représentation des classes, le concept peut être représenté par plusieurs termes comme il peut être représenté par un seul terme dit label qui sera relié par d'autres termes.

### 2.3.2 Relation :

Relie les concepts entre eux et traduit les associations sémantiquement, généralement les relations d'ontologie sont binaires telle que le premier argument appelé domaine et le deuxième appelé co-domaine .

### 2.3.3 Fonctions :

Constituent des cas particuliers de relations, dans laquelle un élément de la relation, (le nième) est défini en fonction des n-1 éléments précédents.

### 2.3.4 Axiomes :

Permettent de modéliser les abstractions vraies du domaine traduites par l'ontologie. D'après Gomez, Les axiomes formels sont utilisés pour vérifier la consistance de l'ontologie. [GOME 99]

### 2.3.5 Instances :

Représentent les éléments singuliers véhiculant les connaissances à propos du domaine.

## 2.4 Types d'ontologies :

Les ontologies peuvent être classifiées selon plusieurs dimensions. Parmi celles-ci, On a :

- Typologie selon le niveau de conceptualisation ;
- Typologie selon le niveau du formalisme utilisé ;
- Typologie selon le niveau de détail de l'ontologie ;
- Typologie selon le niveau de complétude.

### 2.4.1 Typologie selon le niveau de conceptualisation :

- **Ontologie de représentation de connaissances :**

Les ontologies de représentation de connaissances décrivent les connaissances nécessaires à la modélisation d'un système de représentation des connaissances.

- **Ontologie haut niveau :**

Ces ontologies décrivent des concepts de très haut niveau. Elles fournissent une structure et un ensemble de concepts généraux sur lesquels les ontologies du domaine (par exemple le domaine médical ou financier ou autres.) pourraient être construites.

- **Ontologie générique :**

Les ontologies génériques ou les méta-ontologies sont des ontologies qui décrivent des connaissances de hautes abstractions. Cela concerne les concepts tels que le temps, l'espace, les événements, etc. qui sont indépendants d'un domaine ou d'un problème particulier. Ce qui les rend réutilisables dans différents domaines.

- **Ontologie de domaine :**

La plupart des ontologies existantes sont des ontologies du domaine. Ces ontologies contiennent des connaissances spécifiques à un domaine particulier. Cependant, elles sont assez génériques pour être utilisées et réutilisées sur différentes applications.

- **Ontologie de tâche :**

Les ontologies de tâche sont des ontologies qui décrivent des connaissances relatives à une tâche générique ou une activité telle que : diagnostic ou planification.

- **Ontologie d'application :**

Les ontologies d'applications sont les ontologies les plus spécifiques. Elles sont conçues pour une application précise dans un domaine particulier. Elles ne sont pas réutilisables. Par exemple, une ontologie développée dans le but de per-

mettre une interopérabilité entre deux systèmes précis, ne pourra être utilisée que dans l'application pour laquelle elle a été créée. [BELA 15]

### 2.4.2 Typologie selon le niveau de formalisation utilisée :

Par rapport au niveau du formalisme de représentation du langage utilisé pour rendre l'ontologie opérationnelle, Uschold et Grüninger proposent une classification comprenant quatre catégories[USCH 96] :

- **Informelles** : l'ontologie est exprimée en langage naturelle. Cela peut permettre de rendre plus compréhensible l'ontologie pour l'utilisateur, mais cela peut rendre plus difficile la vérification de l'absence de redondances ou de contradiction
- **Semi-informelles** : l'ontologie est exprimée dans une forme restreinte et structurée de la langue naturelle ; cela permet d'augmenter la clarté de l'ontologie tout en réduisant l'ambiguïté.
- **Semi-formelles** : l'ontologie est exprimée dans un langage artificiel définit formellement.
- **Formelles** : l'ontologie est exprimée dans un langage artificiel disposant d'une sémantique formelle, permettant de prouver des propriétés de cette ontologie. L'intérêt d'une ontologie formelle est la possibilité d'effectuer des vérifications sur l'ontologie : complétude, non redondance, consistance, cohérence,...

### 2.4.3 Typologie selon le niveau de détail de l'ontologie :

Pour le niveau de détail utilisé lors de la conceptualisation de l'ontologie en fonction de l'objectif opérationnel envisagé pour l'ontologie, deux catégories au moins peuvent être identifiées[PSYC 03] :

- **Granularité fine** : Ce niveau correspond à des ontologies très détaillées, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche.[HACI 14 ]
- **Granularité large** : Ce niveau correspond à des vocabulaires moins détaillés. Par exemple les scénarios d'utilisation spécifique ou les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente. Les ontologies de haut niveau possèdent une granularité large, compte tenu que les concepts qu'elles traduisent sont normalement raffinés ultérieurement dans d'autres ontologies de domaine ou d'application.[HACI 14 ]

#### 2.4.4 Typologie selon le niveau de complétude :

Selon[BACH 00], On peut définir trois niveaux de complétude :

- **Niveau sémantique** : Tous les concepts, caractérisés par un terme/libellé, doivent respecter les quatre principes différentiels :
  - Communauté avec l'ancêtre.
  - Différence (spécification) par rapport à l'ancêtre.
  - Communauté avec les concepts frères (situés au même niveau).
  - Différence par rapport aux concepts frères.

Ces principes correspondent à l'engagement sémantique et assurent que chaque concept aura un sens univoque et non contextuel associé. Deux concepts sont identiques si l'interprétation du terme/libellé à travers les quatre principes différentiels aboutit à un sens équivalent.

- **Niveau-Référentiel** : Les concepts référentiels ou formels, se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. L'engagement ontologique spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle. Deux concepts formels seront identiques s'ils possèdent la même extension.
- **Niveau-Opérationnel** : Les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des interfaces ou engagement computationnel.

## 2.5 Cycle de vie d'une ontologie :

Les ontologies sont considérées comme des composants logiciels dans les systèmes et leurs développements s'appuyaient sur le même principe appliqué au génie logiciel d'où ils possèdent un cycle de vie. Les activités liées aux ontologies sont d'une part des activités de gestion de projet :

- Planification ;
- Contrôle ;
- Assurance qualité.

Et d'autre part, des activités de développement :

- Spécification ;
- Conceptualisation ;
- Formalisation.

S'y ajoutent des activités transversales de support telles que l'évaluation, la documentation et la gestion de la configuration.[[BANE 07](#)] Ainsi, son cycle s'articule autour de :

- **Etape de détection et de spécification des besoins** : Permet de définir le domaine de connaissance ;
- **Etape de conception** : Subdivisée en trois étapes : Normalisation, Formalisation et opérationnalisation, elle permet d'aboutir à un modèle informel, donc sémantiquement ambiguë et généralement exprimé en langage naturel. Cette étape, consiste, à partir des données brutes, à dégager les concepts et les relations entre ces concepts ;
- Etape de déploiement et de diffusion ;
- Etape d'utilisation ;
- Etape d'évaluation ;
- Etape consacrée à l'évolution et à la maintenance.

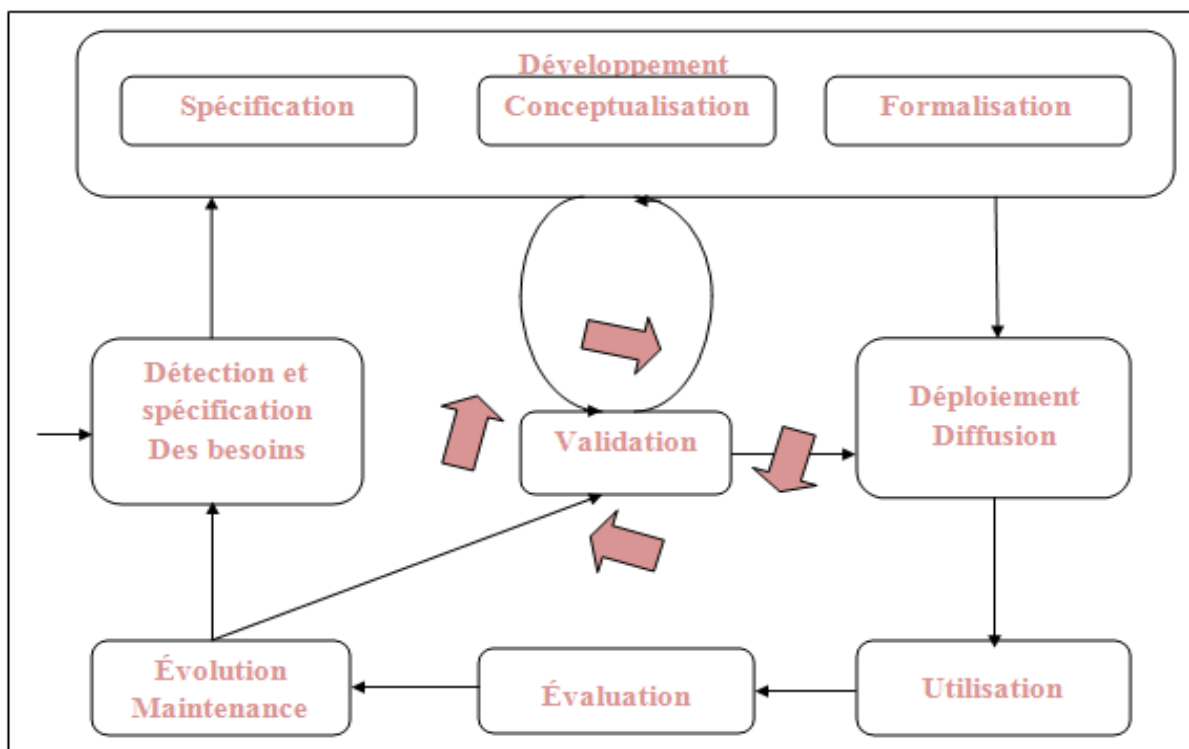


FIGURE 2.1 – Cycle de vie d’une ontologie [BANE 07]

## 2.6 Processus de construction d’une ontologie :

L’activité de construction est la partie centrale du cycle de vie d’une ontologie. Le processus de leur construction est une collaboration qui réunit des experts du domaine de connaissance, des ingénieurs de la connaissance, voire les futurs utilisateurs de l’ontologie. Cette collaboration ne peut être fructueuse que si les objectifs du processus ont été clairement définis.

Voilà les différentes étapes permettant de passer des données brutes à l’ontologie opérationnelle comme le montre la figure ci-dessous :

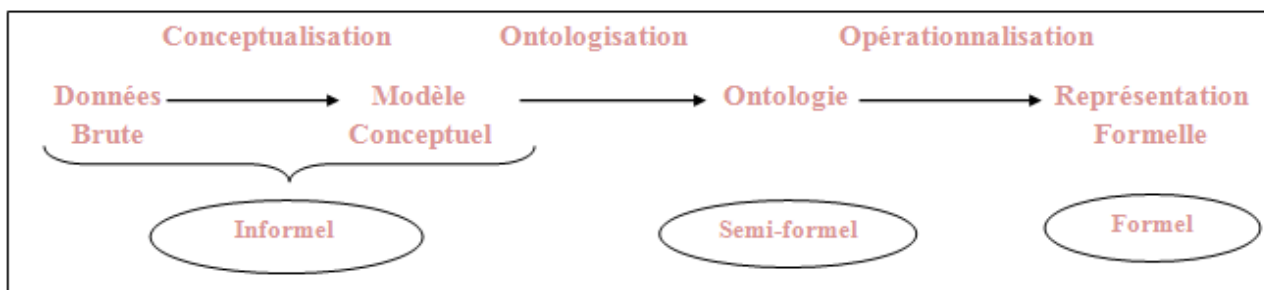


FIGURE 2.2 – Processus général de construction d’une ontologie [BELA 15]



### 2.6.1 Spécification (évaluation des besoins) :

Le développement d'une ontologie commence par la définition du domaine et portée de celle-ci. Cela est basé sur la réponse à certaines questions : Quel est le domaine que l'ontologie va couvrir ? À quoi cette ontologie va servir ? À quels types de questions les informations de l'ontologie doivent fournir des réponses ? Qui va utiliser et maintenir l'ontologie ?, etc. Les réponses à ces questions peuvent changer durant le processus de développement de l'ontologie, mais à chaque étape, elles permettent de limiter la portée du modèle. L'une des solutions qui permet de déterminer la portée d'une ontologie consiste à définir ou planifier une liste de questions auxquelles une base de connaissance, basée sur l'ontologie, doit être capable de répondre. [GHER 17]

### 2.6.2 Conceptualisation :

Cette étape de conceptualisation, permet d'aboutir à un modèle informel, donc sémantiquement ambiguë et généralement exprimé en langage naturel. Cette étape, consiste, à partir des données brutes, à dégager les concepts et les relations entre ces concepts permettant de décrire de manière informelle les entités cognitives du domaine, ainsi, le modèle obtenu consiste en un ensemble de termes désignant les entités du domaine de connaissances (concepts, relations, propriétés des concepts et des relations,.....)[BELA 15]

### 2.6.3 Ontologisation :

L'ontologisation consiste en une formalisation partielle, sans perte d'information, du modèle conceptuel obtenu dans l'étape précédente. Ce qui permet de faciliter sa représentation ultérieure dans un langage complètement formel et opérationnel. Le modèle obtenu est souvent qualifié de semi-formel. [BELA 15]

### 2.6.4 Opérationnalisation :

L'opérationnalisation de l'ontologie consiste à la spécification informatique des opérations applicables aux concepts dans un langage opérationnel. L'utilisation opérationnelle d'une ontologie suppose sa représentation dans un langage formel mais aussi opérationnel, offrant des mécanismes de raisonnements adaptés aux manipulations de connaissances envisagées. Cette étape devra donc intégrer des outils permettant d'opérationnaliser l'ontologie. [BELA 15]

## 2.7 Méthodes et méthodologies de construction d'ontologies :

### 2.7.1 Stratégies de construction d'une ontologie :

Il existe trois stratégies de construction d'une ontologie :

**a- Ascendante** : On commence par les concepts les plus spécifiques aux concepts plus génériques.

**b- Descendante** : On commence par les concepts les plus génériques aux concepts plus spécifiques.

**c- Mixte** : On peut utiliser les deux stratégies, on commence par un concept connu et étendre vers un concept générique ou spécifique.

### 2.7.2 Méthodologies de construction :

Il existe plusieurs méthodes de construction d'ontologie, parmi les quels on a :

**a- Méthodologie d'Uschold et Grüninger**[USCH 96]

Cette méthode repose sur les points suivants :

- Identifier l'objectif souhaité et spécifier le domaine concerné ;
- Construire l'ontologie et pour cela définir les concepts, les relations clés et produire des définitions textuelles précises et non ambiguës de ces concepts ;
- Evaluer le résultat ;
- Documenter le modèle en éditant des recommandations précises pour chaque étape .

**b-La méthode Methontology**[FERN 97], elle est utilisée pour :

- Construire le glossaire des termes qui seront inclus dans l'ontologie, préciser leurs définitions en langage naturel, identifier leurs synonymes et leurs acronymes ;
- Construire des taxonomies de concepts ;
- Construire des diagrammes de relations binaires ;
- Construire le dictionnaire de concepts qui inclut, pour chaque concept, ses attributs d'instance, ses attributs de classe et ses relations ;
- Décrire en détail chaque relation binaire ;
- Décrire en détail chaque attribut d'instance ;
- Décrire en détail chaque attribut de classe ;
- Décrire en détail chaque constante ;
- Décrire les axiomes formels ;

- Décrire les règles utilisées pour contraindre le contrôle et pour inférer des valeurs aux attributs.

**c- Méthodologie de Guarino et Welty :** Cette méthode est une étape qui permet la vérification et la correction d'une structure ontologique construite un peu anarchiquement[WELT 01].

**d- Méthode ARCHONTE :**(ARCHitecture for ONTological Elaborating) elle est proposée par Bruno Bachimont[BACH 00], elle consiste à :

- Choisir les termes pertinents du domaine et normaliser leur sens en précisant les relations de similarités et de différences que chaque concept entretient avec ses concepts frères et son concept père ;
- Formaliser les connaissances, c'est-à-dire ajouter éventuellement des propriétés et des axiomes à des concepts ;
- Opérationnaliser dans un langage de représentation des connaissances.

## 2.8 Langages d'ontologie :

L'ontologie pour qu'elle soit exploitée et partagée par un grand nombre d'utilisateurs, elle doit être exprimée dans un langage qui peut être utilisé par les applications et les plateformes et doit répondre aux exigences des utilisateurs potentiels de cette ontologie

### 2.8.1 Langages traditionnels :

Au début des années 90 les langages de représentation des ontologies sont fondées sur l'intelligence artificielle , il sont basés sur le formalisme de représentation des connaissances suivantes :

- La logique du premier ordre tel que KIF .
- Les frames combinées avec la logique du premier ordre tel qu'Ontolingua, OCML et Flogic .
- La logique de description telle que Loom.[KHAL 09]

### 2.8.2 Web standard ou basé sur xml :

Il existe plusieurs langages qui ont été utilisés pour la description des ontologies. Ils regroupent le langage de représentation des hiérarchies XML (eXtensible Markup Language) , le langage RDF (Resource Description Framework), le langage DAML+OIL (Darpa Modeling Language of Ontology+ Ontology Inference Layer)

et le langage OWL (OntologyWebLanguage). Ces langages offrent différents niveaux d'expressivité.

**a- XML**[BELA 15]

Le langage XML possède une syntaxe pour la description des documents semi-structurés. Il ne permet pas d'imposer des contraintes sémantiques à la signification des documents décrits. Donc XML ne permet pas une interprétation sémantique des données. XML Schéma permet de définir les balises ainsi que l'agencement de ces balises autorisé pour définir la validité d'un document XML.

**b- RDF et RDFs :**

- **RDF** : « Resource Description Framework », est un langage basé sur la syntaxe de XML permet de décrire les objets appelés aussi ressources et de les relier entre eux sur le web . RDF permet de définir, sous forme de graphes de triplets, des données ou des métadonnées.

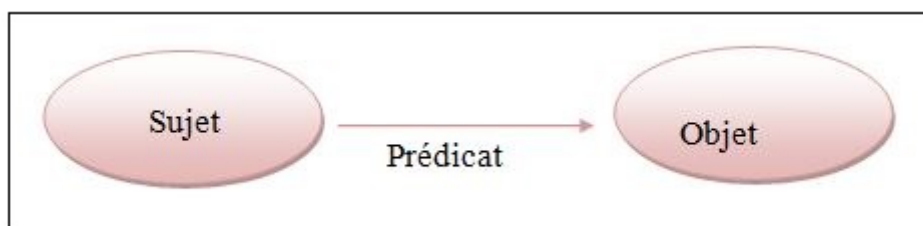


FIGURE 2.3 – Triplet RDF[BENA 17]

- Le RDFs (RDF schéma) est l'extension de RDF incluant des classes, sous-classes, propriétés, sous-propriétés et des règles d'héritage de propriétés.

RDFs propose les caractéristiques suivantes :

- Organisation des objets en classes et les propriétés binaires ;
- Description des sous-classes et les relations de sous-propriétés ;
- Restriction de domaine et d'étendue au niveau de propriétés ;
- RDF et RDFs fournissent les langages de base pour le web sémantique.

**c- DAML+OIL** : est la fusion de deux langages de représentation des connaissances DAML et OIL basés essentiellement sur les logiques de descriptions et qui ont été proposés comme langage de description d'ontologies.

Le but de DAML+OIL est d'étendre RDFS en lui ajoutant des primitives plus expressives pour la définition des classes et des propriétés d'une ontologie. Parmi

ces extensions l'intersection (`daml :intersectionOf`), l'union (`daml :unionOf`) et la négation (`daml :complementOf`), etc... [ABDE 07]

**d- OWL (Ontology Web Language) :**

Les RDFs sont limités aux hiérarchies des sous-classes et de sous propriétés avec restriction du domaine, il existe plusieurs cas particuliers d'utilisation du web sémantique qui nécessitent une grande expressivité. Ce type d'extension comprend :

La disjonction, combinaison booléenne des classes, les restrictions de cardinalités et les caractéristiques des propriétés.

OWL a été élaboré comme nouveau langage d'ontologies .Il a été créé en 2001 par le W3C, il hérite du langage DAML+OIL, il repose sur RDFs. En plus, OWL est doté de trois sous langages offrant des capacités d'expression croissantes :

- OWL Lite : est un sous langage de OWL DL, il définit les concepts simples de OWL. La cardinalité est limitée à 0 ou 1.
- OWL Full : autorise le mélange arbitraire avec le schéma RDF. Il est plus complet que OWL Lite et OWL DL et il est utilisé pour la transformation des concepts du diagramme de classe UML.[MACA 02 ]
- OWL DL : est un sous langage de OWL Full et se base sur la logique descriptive. Il défini des concepts plus complexes comme le complément, l'intersection, l'union, etc.

OWL est un facteur important parce qu'il permet de :

- Vérifier la cohérence d'une ontologie et d'une connaissance ;
- Vérifier la présence de relations non voulues entre classes ;
- Classifier automatiquement les instances en classes.

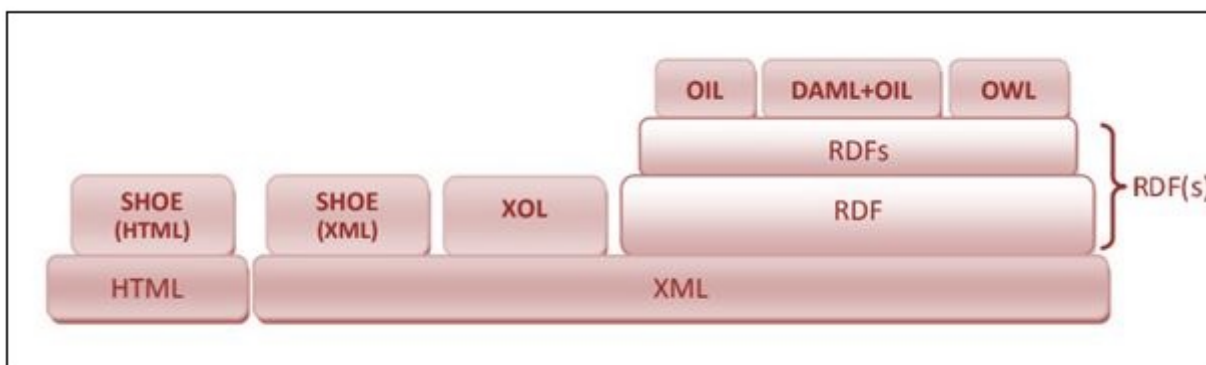


FIGURE 2.4 – Les langages d'exploitation des ontologies[GOME 99]

### 2.8.3 Langages d'interrogation des ontologies :

#### Utilisation du protocole de requête SPARQL :

Une fois l'ontologie construite et sauvegardée dans un fichier OWL, il faut utiliser un protocole de requête spécifique au langage RDF construit sur des graphes de données.

Le SPARQL (prononcer « sparkle », acronyme de SPARQL Protocol And RDF Query Language), recommandé depuis le 15 Janvier 2008 par le W3C. SPARQL est un langage de requête et un protocole qui permettra de rechercher, d'ajouter, de modifier ou de supprimer des données RDF/RDFS disponible à travers Internet.

SPARQL s'apparente à SQL qui comme SQL accède aux base de données via une langage de requête, mais SPARQL accède aux ressources présentes sur le Web. [GHER 17]

### 2.8.4 Approches du raisonnement :

À partir de la technique utilisée dans le développement des algorithmes de décision pour le problème d'inférence, il existe deux groupes d'algorithmes. Le premier est appelé algorithmes structurels, ce sont des algorithmes qui comparent la structure syntaxique des concepts. Et le deuxième groupe qui est appelé algorithmes de tableaux, algorithmes donnés la première fois par Schmidt-Schauß et Smolka[SCHM 89], et qui deviennent réellement un outil principal pour les problèmes de satisfaisabilité et de subsumption de concepts dans les Logiques de Descriptions (LDs) . [BOUA 16]

#### a-Algorithmes de subsumption structurelle :

Les algorithmes du calcul de subsumption sont basés sur la comparaison structurelle entre les expressions de concepts. L'idée de cette approche est que si deux expressions de concept sont faites de sous-expressions, alors elles peuvent être comparées séparément, en comparant la sous-expression d'un concept avec toutes celles de l'autre. L'algorithme s'exécute en deux phases : premièrement, les concepts sont réécrits sous une forme normale (déplier les concepts et factoriser les rôles), et ensuite leurs structures sont comparées : Toutes les conjonctions emboîtées sont égalisées, c.-à-d

$$A \cap (B \cap C) \iff A \cap B \cap C$$

Toutes les conjonctions de quantifications universelles sont factorisées, c.-à-d

$$\forall P.C \cap \forall P.D \iff \forall P.(C \cap D)$$

Les concepts réécrits sont logiquement équivalents aux précédents, donc la subsomption est préservée par cette transformation. Soient

$$C = C1 \cap C2 \cap \dots \cap C_m \text{ et } D = D1 \cap D2 \cap \dots \cap D_n$$

alors D subsume C si et seulement si pour chaque  $D_i$ , il existe un  $C_j$  avec :

- Si  $D_i$  est un concept atomique, ou est un concept de forme  $\exists P$ , alors  $D_i = C_j$ .
- Si  $D_i$  est un concept de la forme  $\forall P.D, C_j = \forall P.C$  (le même rôle atomique P), alors  $C \subseteq D$

Ces algorithmes de comparaison structurelle sont très efficaces (polynomiaux) pour des langages de LD très peu expressifs .

Un autre avantage est son emploi dans les inférences non-standard. Pour Les langages de LD étant plus expressifs, ils ne peuvent pas détecter toutes les relations de subsomption et d'instance existantes.

Autrement dit, ces algorithmes sont incomplets par rapport aux sémantiques de langage de la logique du première-ordre, si le langage permet les combinaisons particulières de constructeurs.

Toutefois, les algorithmes structurels peuvent être améliorés en tenant compte de combinaisons particulières de constructeurs, mais la difficulté est de prouver la complétude de ces algorithmes (c.-à-d. prouver que des comparaisons ont été considérées pour capturer toutes les interactions possibles). [AMED 97]

**b- Algorithmes de tableaux :**

Cette technique de raisonnement est basée sur les calculs de tableaux pour la logique des prédicats du premier-ordre. Son rôle principal est l'adaptation de calculs aux analyses de la complexité. Les structures de tableaux obtenues en raisonnant avec un langage de LD donné sont soigneusement analysées, et les vérifications redondantes dans les tableaux sont éliminées afin de donner une limite supérieure stricte sur la complexité de la méthode. L'idée principale du calcul de tableau est de vérifier si une formule donnée F est une conséquence logique d'une théorie donnée T. On essaye de construire, avec les règles de propagation convenables, le modèle le plus générique de T où F est faux. Si le modèle est construit avec succès, alors la réponse est NON (parce que F n'est pas un conséquence logique de T, donc la méthode est complète) ; si le modèle construit n'est pas un succès, alors la réponse est OUI (parce qu'il n'existe pas un modèle de T avec F faux, donc F est réellement une conséquence logique de T et la méthode est correcte). Les règles de propagation viennent directement de la sémantique de constructeurs. [BOUA 16]

## 2.9 Outils de développement d'ontologie

Une ontologie doit être développée par un outil qui permet de documenter cette ontologie et avoir un support graphique pour la construire pur cela on présentera quelques outils qui aident l'utilisateur à créer une ontologie et l'exporter dans un langage informatisé.

### 2.9.1 PROTÉGÉ

(développé par le Stanford Medical Informatics de l'Université de Stanford) fournit un environnement graphique permettant l'édition, la visualisation et le contrôle (vérification des contraintes) d'ontologies. Le modèle de représentation de connaissances de PROTÉGÉ, est issu du modèle des frames. Ce dernier contient des classes (pour modéliser les concepts), des slots (pour modéliser les attributs des concepts) et des facettes (pour définir les valeurs des propriétés et des contraintes sur ces valeurs), ainsi que des instances des classes. PROTÉGÉ introduit la notion de méta-classe, dont les instances sont des classes. L'interface très complète ainsi que l'architecture logicielle extensible permettant l'insertion de plusieurs plug-ins offrant de nouvelles fonctionnalités, notamment des pluggins pour gérer les représentations sous forme graphique.[[KHAL 09](#)]

### 2.9.2 ODE (ONTOLOGY DESIGN ENVIRONMENT)

C'est un outil de construction des ontologies au niveau connaissances (Méthodologie METHONTOLOGY). L'objectif de l'ODE est de soutenir l'onologue tout au long du développement de l'ontologie, de la spécification des exigences, de l'acquisition des connaissances et la conceptualisation, à la mise en œuvre, avec autant d'intégration et d'évaluation que possible. [[MARI 99](#)]

### 2.9.3 OntoEdit

ONTOEDIT est un environnement d'ingénierie ontologique mis au point par l'institut AIFB de l'université de Karlsruhe et qui est maintenant commercialisé par la société Ontoprise GmbH. Cet outil est fondé sur un processus de développement d'ontologies suivant les différentes étapes de la méthode de construction ONTO-KNOWLEDGE et met à disposition de l'utilisateur plusieurs vues graphiques correspondant aux différentes phases de conception de l'ontologie. [[KHAL 09](#)]



### 2.9.4 WebOde

WebOde13 est une plateforme en ligne développée par le groupe Ontological Engineering du département d'intelligence artificielle de la faculté d'informatique de l'université polytechnique de Madrid. C'est un éditeur qui assurait le support de Methontology. L'éditeur d'ontologie de WebODE permet d'éditer et de naviguer dans les ontologies WebODE et il se base sur des formulaires HTML et des applets Java [BENA 17]

### 2.9.5 DoE

DoE (Differential Ontologies Editor) a été développé à l'Institut National de l'Audiovisuel par R. Troncy et A. Isaac en 2002 [ISSA 05]. DOE est un éditeur qui offre des interfaces de création, modification et suppression de concepts et de relations, une représentation graphique de l'arbre ontologique, et des fonctionnalités de recherche et de navigation dans la structure créée. L'ontologie est documentée par des définitions encyclopédiques avec des synonymes et les principes différentiels en plusieurs langues [BANE 07].

### 2.9.6 Framwork jena

Apache Jena (ou Jena en bref) est une plateforme Java gratuite et open source pour la création des applications Web sémantique. JENA est composé de différentes API interagissant ensemble pour traiter les données dans des documents RDF, RDFS, OWL et SPARQL. Il fournit un moteur d'inférences permettant des raisonnements sur les ontologies. JENA est maintenant sous Apache Software Licence. [BENA 17]

## 2.10 UML – OWL

### 2.10.1 Le Langage UML

UML est un langage ou formalisme de modélisation graphique. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». UML représente un moyen pour spécifier et représenter les composantes d'un système informatique. Parmi les objectifs d'UML : être indépendant des langages de programmation et être adapté à toutes les phases de développement. [MACA 02 ]

### 2.10.2 Les règles de passage d'un modele UML vers OWL

#### 2.10.2.1 Le Concept de classe

Ce tableau exprime la notion de classe en UML, le graphe et le code correspondant à cette classe en OWL.

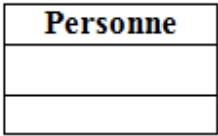
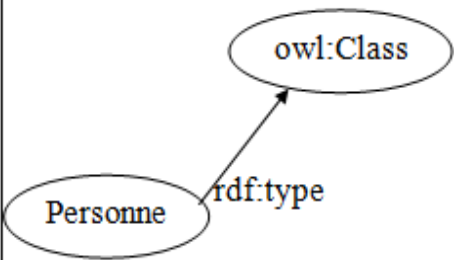
UML	Graphe OWL	Code OWL
		<pre>&lt;owl:Class rdf:ID= "Personne"/&gt;</pre>

TABLE 2.1 – Notion de classe en UML, graphe et code

### 2.10.2.2 Le Concept de propriété

En OWL il y a deux types de propriétés : les propriétés d'objet (owl:ObjectProperty) ou les associations entre classes et les propriétés d'attribut (owl:DatatypeProperty). On peut exprimer la réciproque d'une propriété. La propriété d'objet et sa réciproque sont équivalentes aux rôles en UML ; de même, on peut exprimer des cardinalités sous forme de contraintes de cardinalité sur les propriétés d'objets en OWL. [MACA 02 ]

**exemple** : « Un auteur connu, crée au moins un Œuvre ».

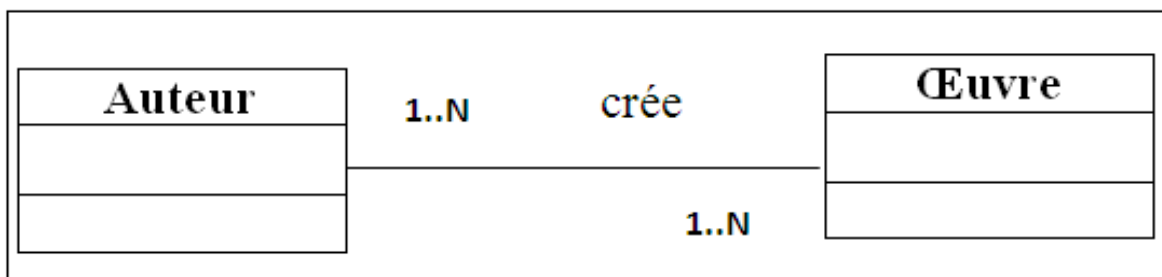


FIGURE 2.5 – Diagramme de Classe UML

- Les classes définies sont : la classe Auteur et la classe Œuvre.
- La propriété d'objet « crée » a pour domaine la classe «Auteur» et pour Image la classe «Œuvre».
- La propriété réciproque de « crée » est «estcréepar».
- La propriété d'attribut de la classe Auteur est « nom ».

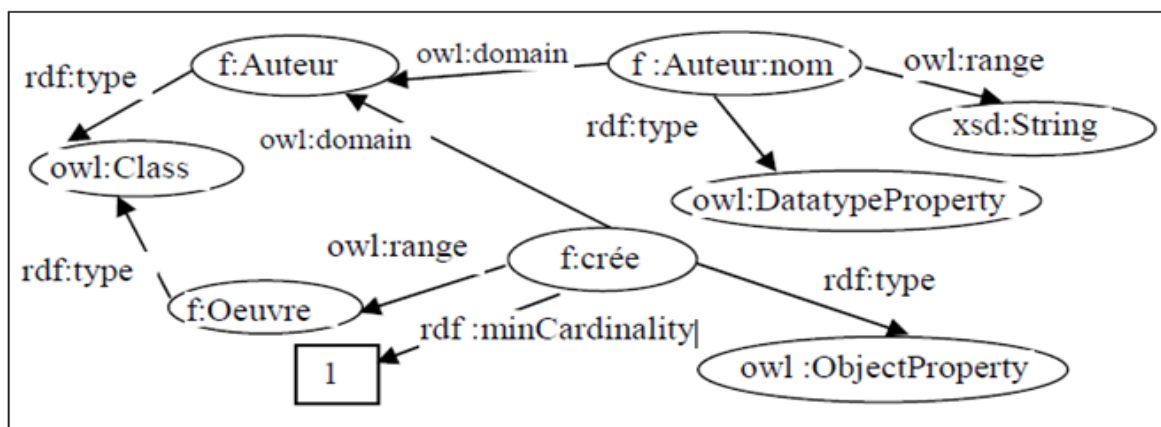


FIGURE 2.6 – Graphe OWL correspondant à l'exemple de propriété OWL. [MACA 02 ]

## 2.11 Conclusion

A travers ce que nous avons présenté dans ce chapitre, il a été mis en évidence que la notion d'ontologie représente un élément primordial pour toute structuration sémantique de données portant sur un domaine particulier. Nous avons tenté de clarifier le principe d'ontologie en présentant certaines définitions. Ensuite, nous avons exposé les composantes d'une ontologie, les différentes classifications, leur cycle de vie, ainsi que leurs méthodologies de construction.

D'autre part, nous avons présenté les différents langages d'ontologie tel que les langages de représentation, les langages d'interrogation, et de raisonnement ainsi les outils de développement d'ontologie et enfin nous avons parlé sur le langage UML et les règles de passage d'un UML à OWL. Au cours de ce projet, nous tentons de contribuer l'ontologie dans notre projet qui est la maintenance du système de refroidissement d'un moteur de voiture en utilisant le raisonnement à base de cas .

Dans le chapitre suivant, nous présentons les aspects conception et réalisation de notre outil de raisonnement à base de cas intégrant les ontologies.

# Chapitre 3

## Modélisation et implémentation du système Onto-master

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>42</b>
<b>3.2</b>	<b>Partie I : Conception et Modélisation</b>	<b>42</b>
<b>3.3</b>	<b>Partie II : Mise en œuvre</b>	<b>49</b>
<b>3.4</b>	<b>Conclusion</b>	<b>65</b>

---

## **3.1 Introduction**

La dernière phase de notre mémoire est consacrée à la modélisation ainsi qu'à la mise en place de la solution logicielle de notre système. Ce chapitre est structuré comme suit : la section une est consacrée à la modélisation et conception de notre système où nous allons présenter les modèles et diagrammes conçus. La deuxième section est dédiée à la description des outils logiciels utilisés lors du développement de notre système à savoir protégé, netbeans, etc., ainsi que les scénarios d'exécution.

## **3.2 Partie I : Conception et Modélisation**

Présentation de notre approche :

### **3.2.1 Domaine d'étude (description du système d'étude)**

Les sections suivantes (3.2.3 – 3.2.5) donnent un aperçu sur le système de refroidissement d'un moteur d'un véhicule qui est le champ de notre étude.

### **3.2.2 Présentation du système de refroidissement**

Un système de refroidissement est essentiel au bon fonctionnement de tous les véhicules. Ses fonctions premières, délivrées grâce à ses différents composants, sont d'assurer la montée en température rapide du moteur et de maintenir une température de fonctionnement ainsi qu'une pression constante pour éviter toute surchauffe.

### 3.2.3 Fonctionnement du système de refroidissement

- Au démarrage, le circuit est réduit afin de favoriser une montée en température rapide du moteur. Il est représenté en rouge sur la figure ci-dessous.

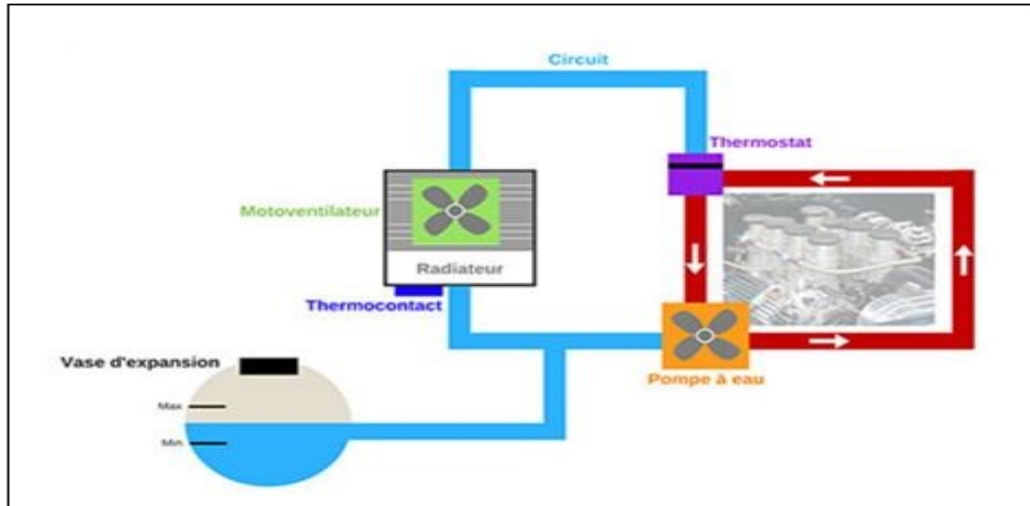


FIGURE 3.1 – Fonctionnement du système de refroidissement(a)

- Lorsque le moteur est chaud, le thermostat s'ouvre ce qui permet au liquide d'accéder à l'autre partie du circuit (ici en bleu) afin d'éviter une surchauffe. La température optimale est de  $88^{\circ}$ .

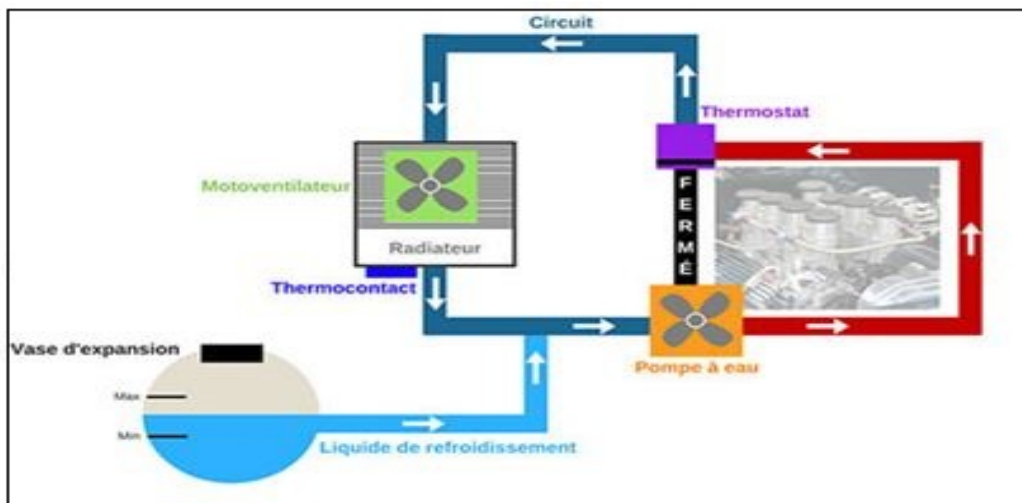


FIGURE 3.2 – Fonctionnement du système de refroidissement(b)

### 3.2.4 Eléments du système

Le système de refroidissement des automobiles est composé de tout un ensemble d'éléments, et notamment :

- Le liquide de refroidissement
- Les durites
- La pompe à eau
- Le thermostat ou calorstat
- Le radiateur
- Le thermocontact
- Motoventilateur
- Le vase d'expansion

### 3.2.5 Rôle des éléments du système

#### - Le liquide de refroidissement :

Le liquide de refroidissement est la base du système. Il a deux fonctions :

- Acheminer les calories jusqu'au radiateur pour les évacuer.
- Refroidir le moteur.

#### - Les durites :

Sont les canalisations transportant le liquide refroidissement, elles sont généralement en métal ou en caoutchouc.

#### -La pompe à eau :

Elle a pour rôle de gérer le débit du liquide dans le circuit de refroidissement. Pour cela la pièce est munie d'une hélice commandée par le moteur.

#### - Le thermostat ou calorstat :

Il se charge d'ouvrir et fermer l'accès au grand circuit de refroidissement. Il s'agit d'une vanne thermique répondant au changement de température du liquide de refroidissement (système de cire qui se dilate selon la chaleur) afin de libérer ou non le passage.

#### - Le radiateur :

Le radiateur situé à l'avant de la voiture a un double objectif :

- Évacuer les calories dans l'air libre
- Refroidir si nécessaire le liquide de refroidissement grâce à l'air qu'il reçoit (le métal garde particulièrement bien la fraîcheur).

#### - Le thermocontact et le motoventilateur :

Lorsque le radiateur reçoit peu ou pas assez d'air, un ventilateur situé devant ce dernier se charge alors d'en apporter. On parle de motoventilateur. Celui-ci



s'active en fonction du thermocontact, une sonde thermostatique placée au niveau du radiateur mesurant la température du liquide de refroidissement à cet endroit-là. Si celle-ci est trop élevée, le groupe motoventilateurs (GMV) s'active alors.

**- Le vase d'expansion :**

Le vase d'expansion est le récipient contenant le liquide de refroidissement. On peut mesurer le niveau grâce à un système classique de minimum et maximum. Selon le débit et la température au sein du circuit de refroidissement, du liquide supplémentaire peut être ajouté.

### 3.2.6 Architecture générale de notre application

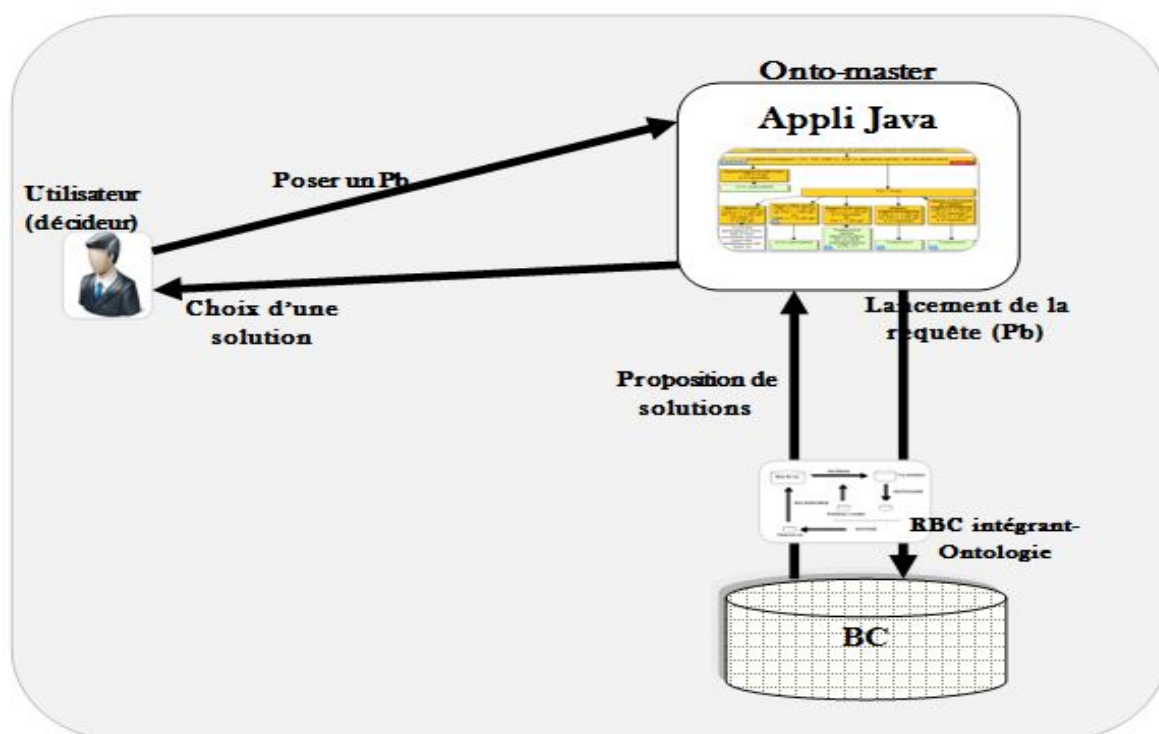


FIGURE 3.3 – Représentation générale de l'application

### 3.2.7 Connaissances

Les connaissances que nous avons considérées au niveau de notre système sont la base de cas, intégrant implicitement trois ontologies : de tâche, de domaine d'application ainsi que celle du domaine du matériel à maintenir. L'exploitation de ces connaissances consiste à établir des relations entre elles.

### 3.2.8 Base de cas

Pour chacun des cas nous considérons la représentation suivante : (problème (la tâche à résoudre, la causes du problème et les symptômes) solution (la méthode de résolution, l'objet concerné par la solution préconisée)). La figure suivante présente le diagramme de classes UML relatif à la modélisation de la base des cas.

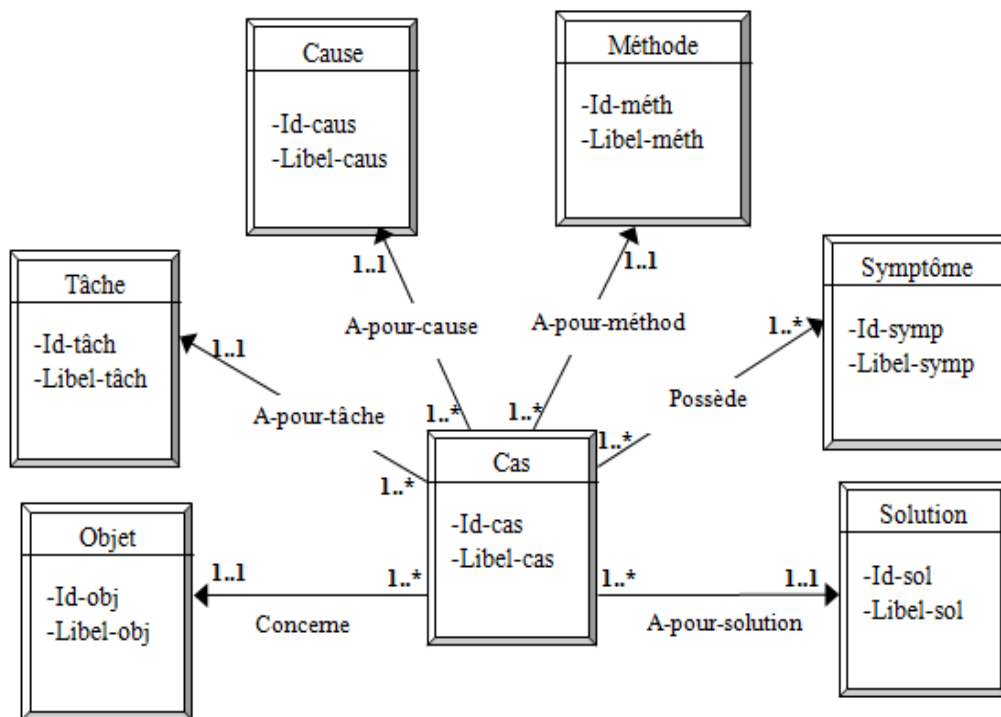


FIGURE 3.4 – Diagramme de classe UML de la base de cas

### 3.2.9 Ontologies

Nous avons déjà cité dans le chapitre 2 différents types d'ontologies, nous allons les exprimer ci-dessous :

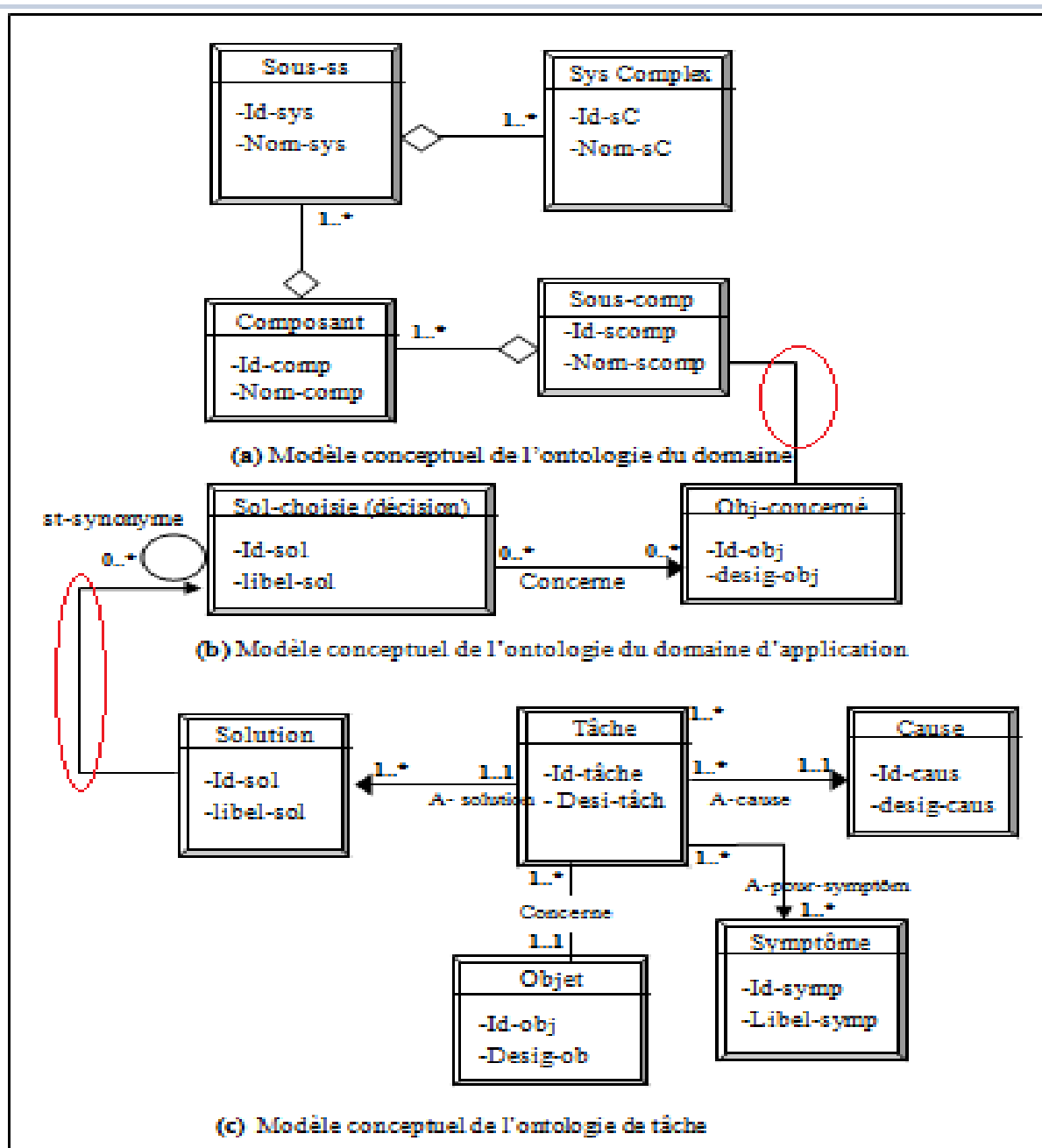


FIGURE 3.5 – Différents modèles conceptuels d'ontologie

- (a) Modèle conceptuel de l'ontologie du domaine.
- (b) Modèle conceptuel de l'ontologie du domaine d'application.
- (c) Modèle conceptuel de l'ontologie de tâche.

### 3.2.10 Diagramme de cas d'utilisation

Le diagramme suivant (figure 3.6), définit clairement les acteurs qui interagissent avec le système.

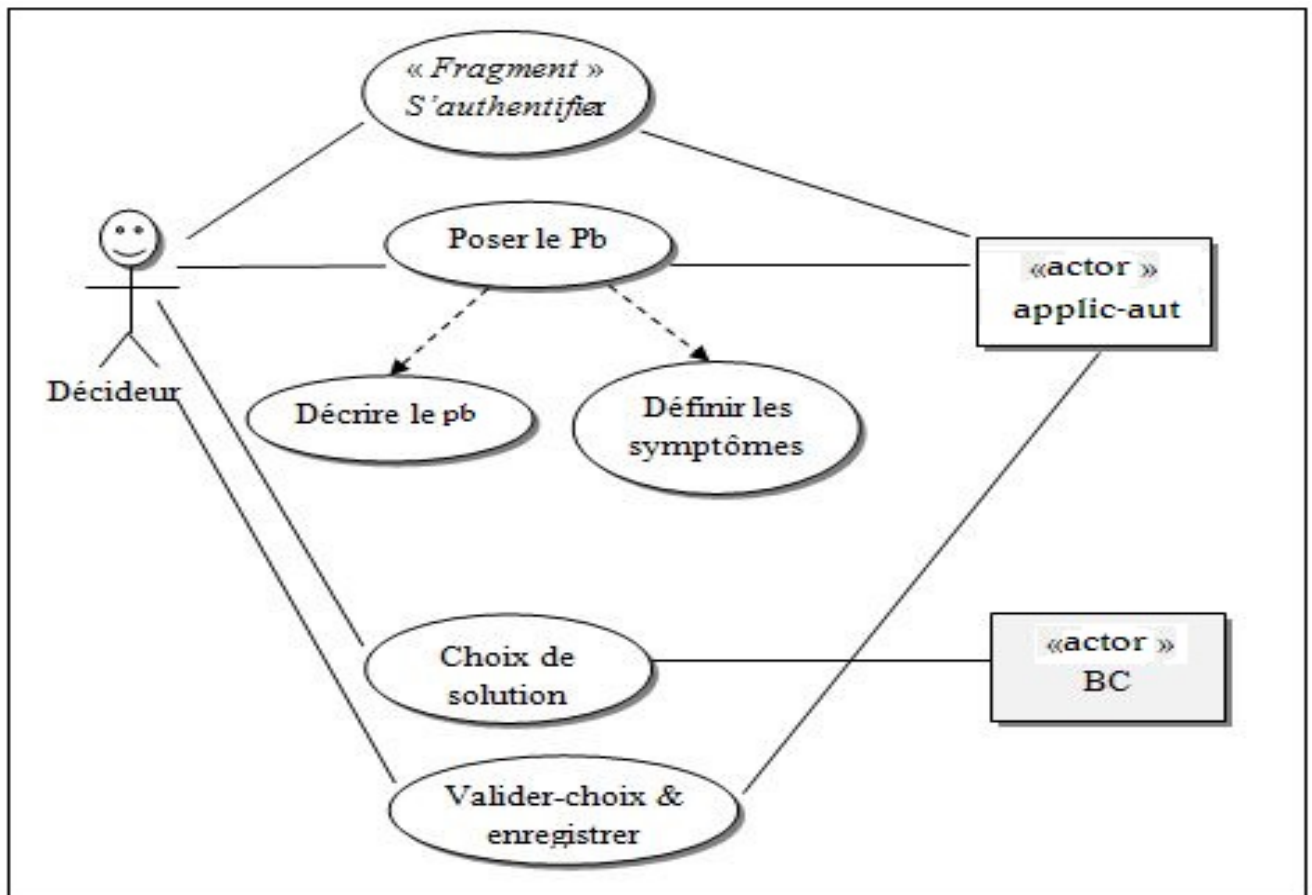


FIGURE 3.6 – Diagramme de cas d'utilisation

## 3.3 Partie II : Mise en œuvre

### 3.3.1 Environnement de programmation

Il existe plusieurs outils et langages pour des applications des ontologies industrielles. Parmi ces outils nous avons utilisé NetBeans IDE et Jena. Et parmi les langages nous avons utilisés JAVA.

#### 3.3.1.1 NetBeans 8.2



NetBeans est un environnement de développement intégré (EDI), placé en open source en juin 2000 sous licence CDDL (Common Développement and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible sous Windows, Linux, ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).

NetBeans constitue par ailleurs une plate-forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate-forme, il s'enrichit à l'aide de plugins.[\[AHME 18\]](#)

### 3.3.1.2 JENA



JENA est un environnement de travail open source en Java, pour la construction d'application web sémantique. JENA permet de manipuler des documents RDF, RDFS, OWL et SPARQL. Il fournit un moteur d'inférences permettant des raisonnements sur les ontologies. JENA est maintenant sous Apache Software Licence. [BELA 15]

La (FIGURE 3.7) nous montre comment nous avons ajouté la bibliothèque jena à notre projet.

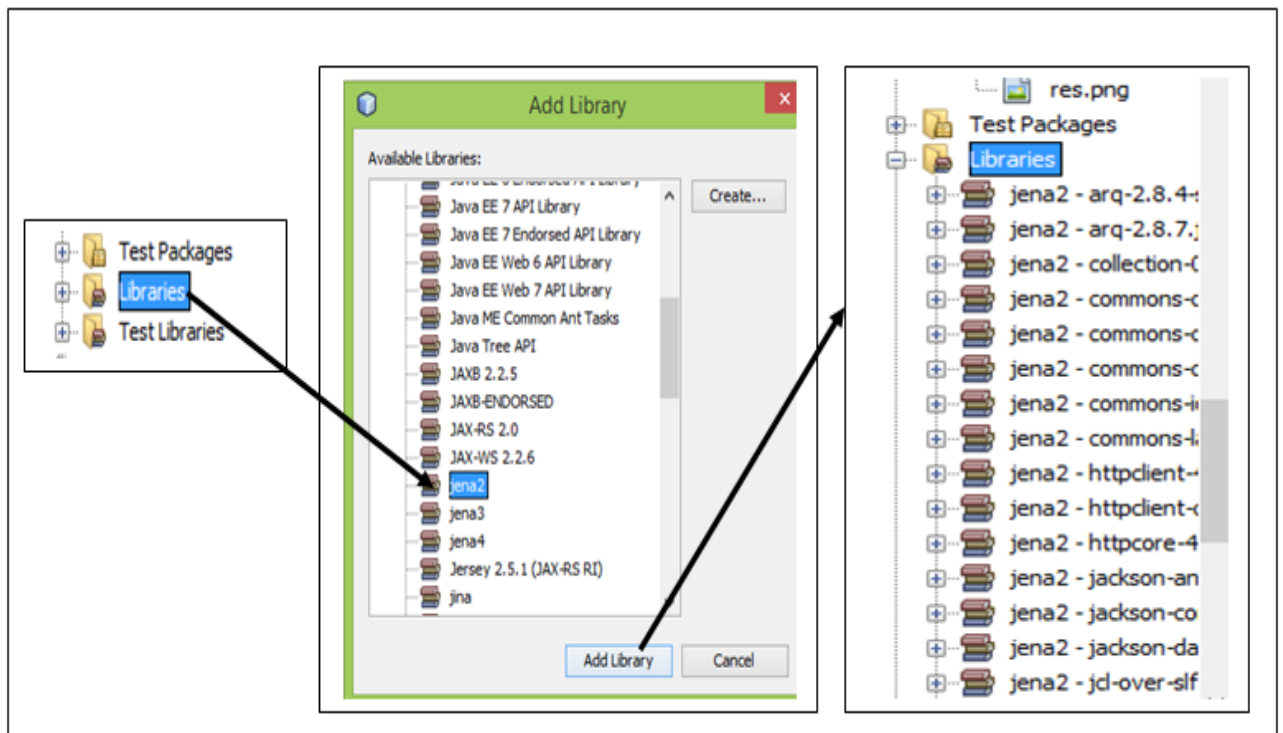


FIGURE 3.7 – Ajout de la bibliothèque API jena à notre projet

### 3.3.1.3 JAVA

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy

Il existe plusieurs IDE (Integrated Development Environment) pour le langage JAVA par exemple Eclipse, JBuilder et NetBeans que nous avons utilisé. [HADJ 12]

### 3.3.1.4 Protégé 4.2



PROTEGE est une interface modulaire, développée au Stanford Medical Informatics de l'Université de Stanford, permettant l'édition, la visualisation, le contrôle, l'extraction à partir de sources textuelles, et la fusion semi-automatique d'ontologies. PROTEGE OWL autorise la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie. De nombreux plugins sont disponibles ou peuvent être ajoutés par l'utilisateur. [AHME 18]

### 3.3.2 Étapes de création de notre ontologie "onto" :

#### 1- Création des classes :

Tout d'abord on a commencé par la création des classes, la FIGURE 3.8 indique ceci :

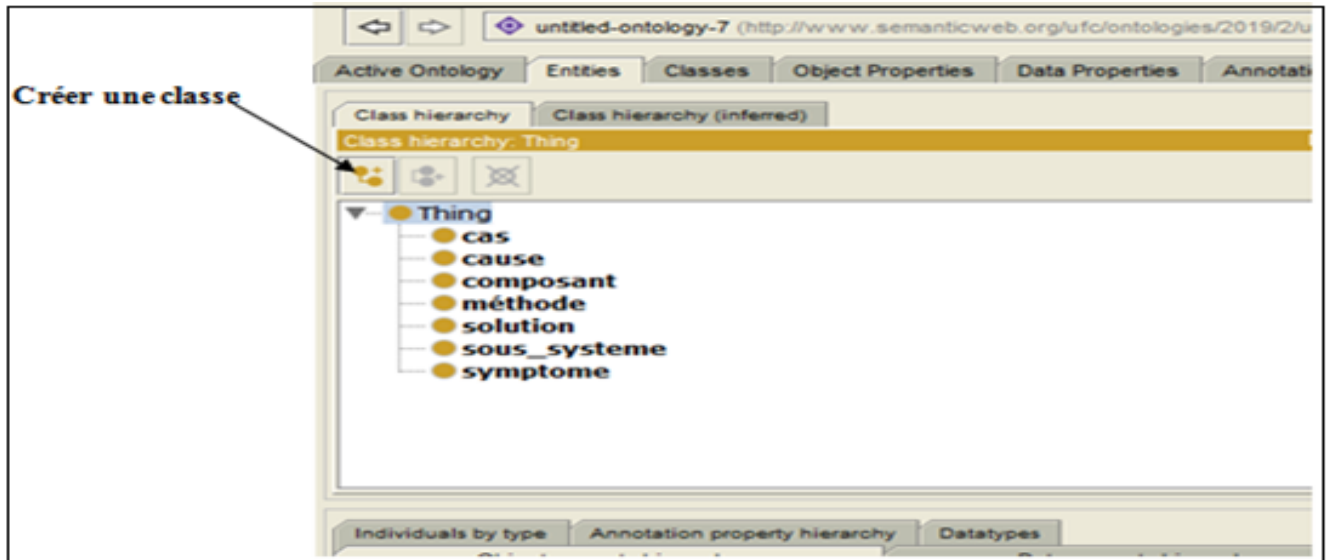


FIGURE 3.8 – Création des classes

#### 2- Création des propriétés :

Après avoir créé les classes, viennent maintenant les liens les reliant, qui se concrétisent par les propriétés "objectProperty".

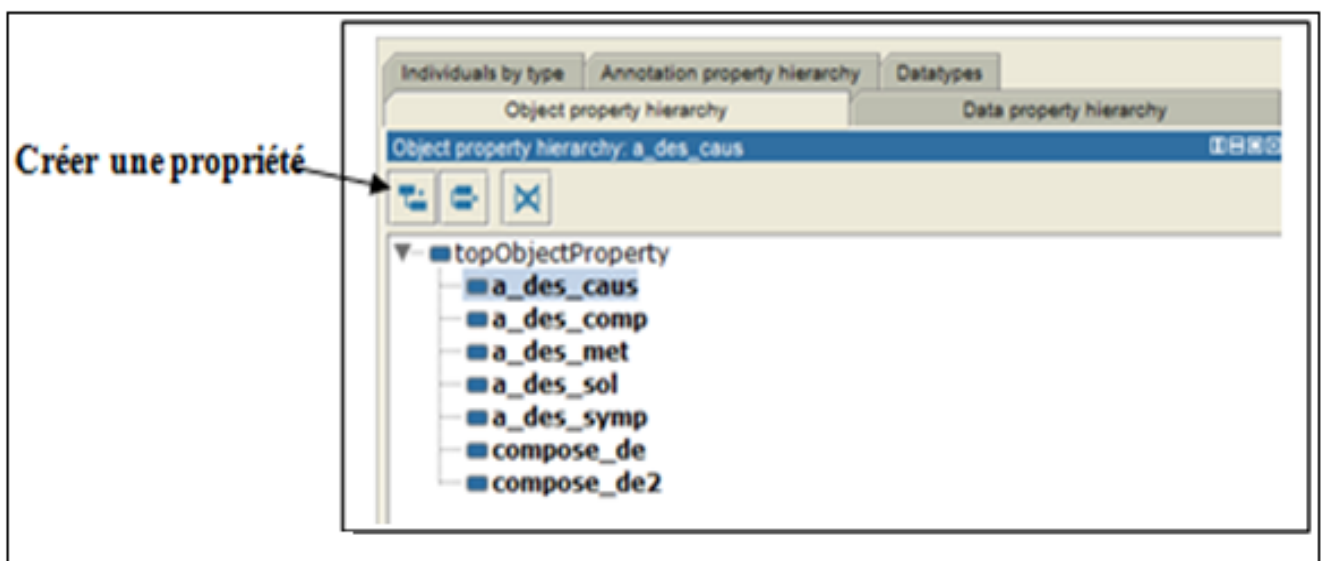


FIGURE 3.9 – Création des propriétés pour les classes



**3- Création des instances :** La FIGURE 3.10 représente la création des individus pour chaque classe.

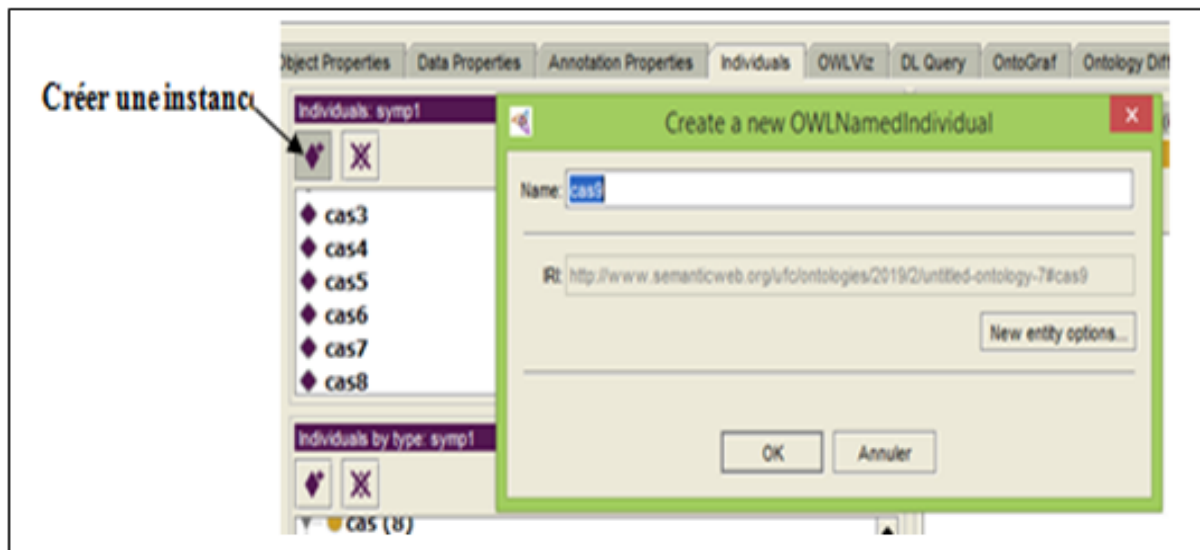


FIGURE 3.10 – Création des individus(instances)

#### 4- Individualisation :

Dans la dernière étape, vient l'attribution des valeurs à chaque instance ce qui est appelé l'individualisation.

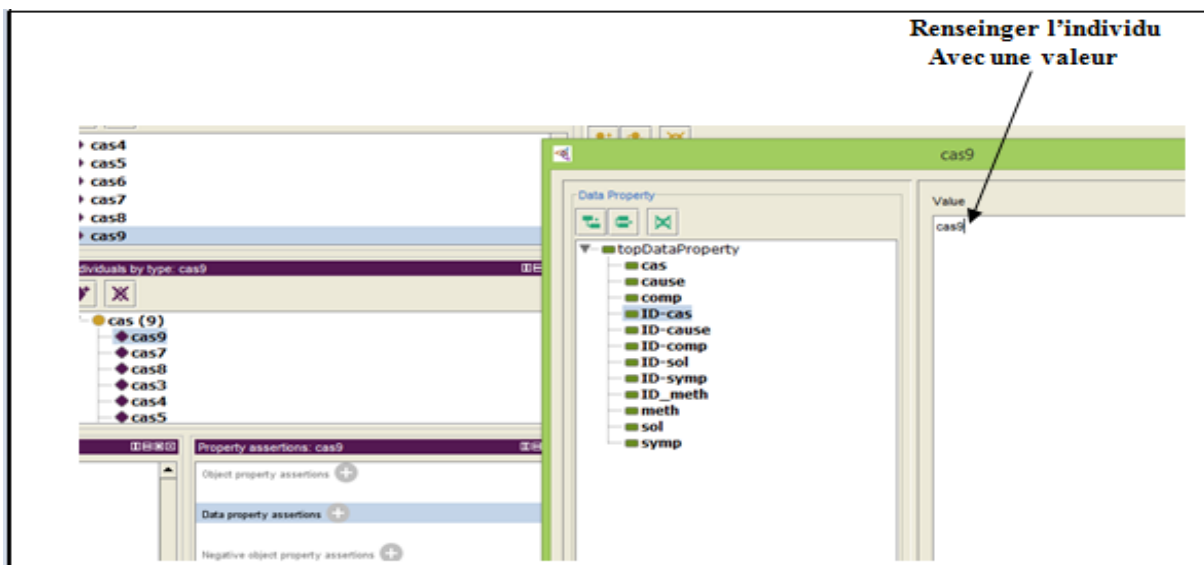


FIGURE 3.11 – Attribution des valeurs aux instances

**Présentation de notre ontologie :**

La figure ci-dessous représente la hiérarchie des concepts de de notre ontologie "onto" avec Protégé OWL.

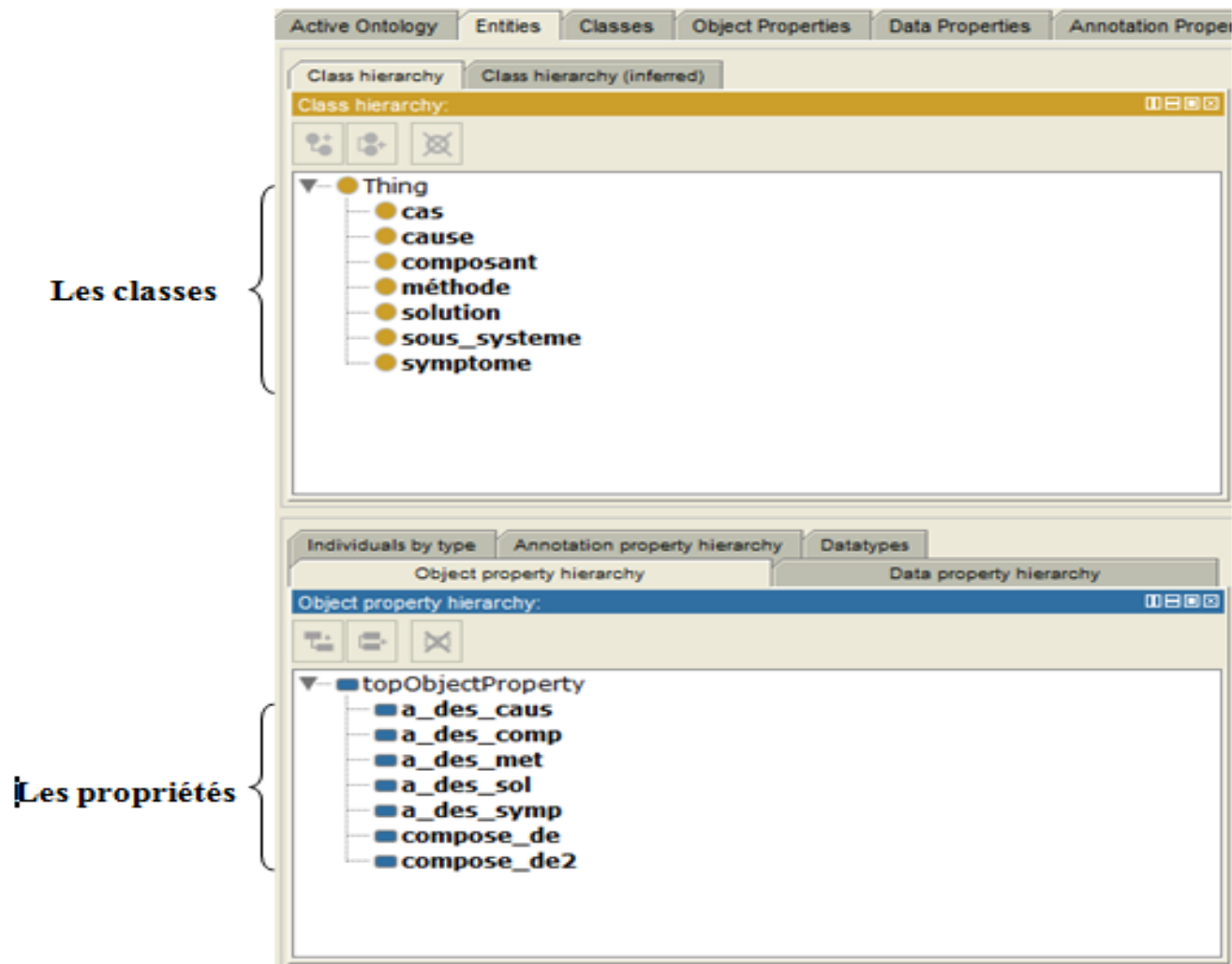


FIGURE 3.12 – Hiérarchie des concepts de l'ontologie "onto" avec Protégé OWL

La FIGURE 3.13 représente graphiquement notre ontologie par l'outil OntoGraph intégré dans protégé.

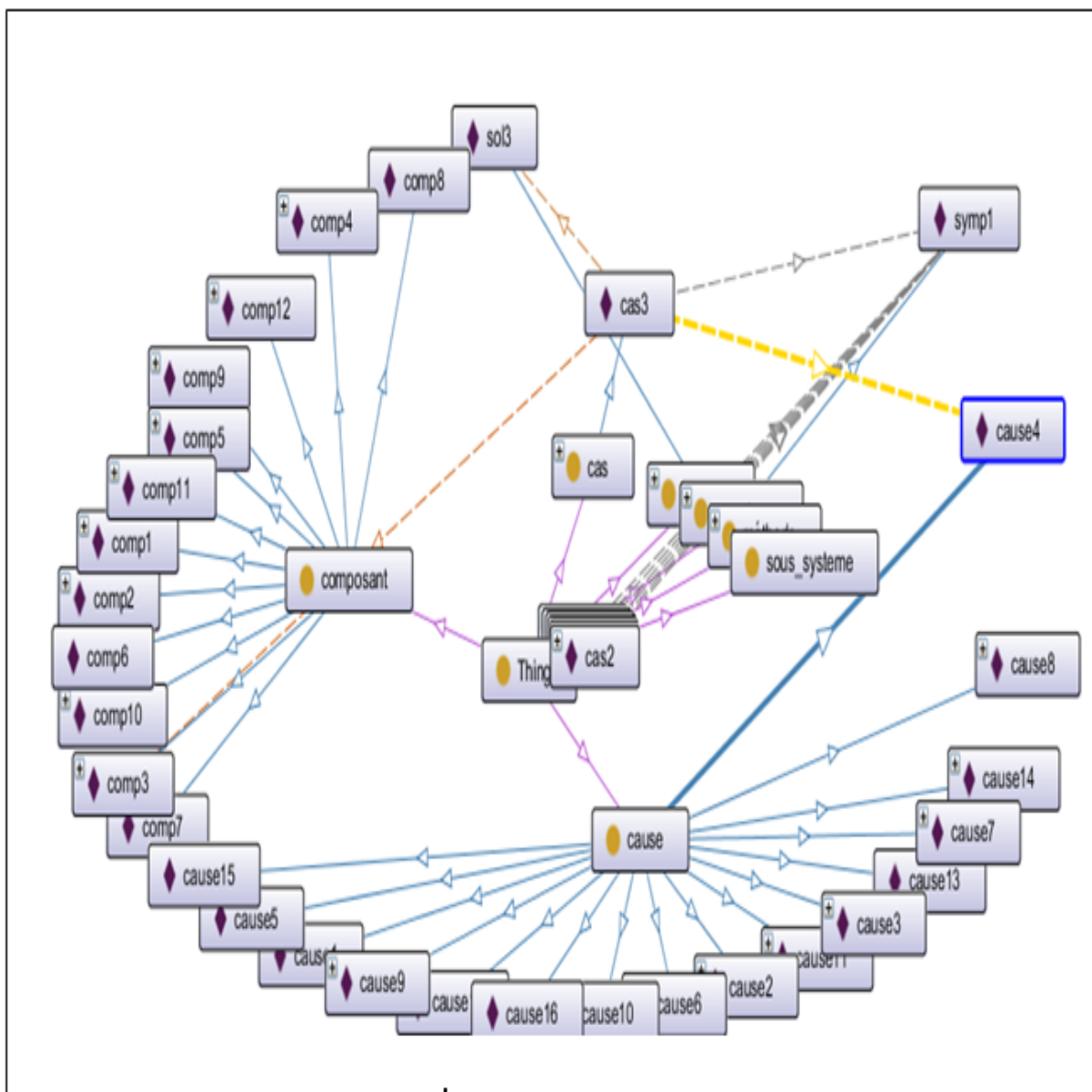


FIGURE 3.13 – Interface OntoGraph

Nous vous montrons dans ce qui suit un extrait du fichier owl de notre ontologie :

```

<!-- http://www.semanticweb.org/ufc/ontologies/2019/2/untitled-ontology-7#cas1 -->

<owl:NamedIndividual rdf:about="#untitled-ontology-7;cas1">
  <rdf:type rdf:resource="#untitled-ontology-7;cas"/>
  <cas>Bouchon du vase d'expansion est faussé</cas>
  <ID-cas>cas1</ID-cas>
  <a_des_caus rdf:resource="#untitled-ontology-7;cause1"/>
  <a_des_comp rdf:resource="#untitled-ontology-7;comp1"/>
  <a_des_sol rdf:resource="#untitled-ontology-7;sol1"/>
  <a_des_symp rdf:resource="#untitled-ontology-7;symp1"/>
</owl:NamedIndividual>

```

FIGURE 3.14 – Ontologie OWL

Et pour pouvoir établir une connexion entre le fichier owl sous protégé et notre application java, nous utilisons la méthode suivante :

```

-----
static void InsertUpdate( String queryString)
UpdateRequest query = UpdateFactory.create(queryString);
UpdateAction.parseExecute(queryString, OpenConnectOWL());
-----

```

### 3.3.3 Quelques Scénarios (guide d'utilisation)

La figure ci-dessous présente l'interface principale de notre application



FIGURE 3.15 – Interface principale

### Identification d'un opérateur :

Dans le but de résoudre un pb lié au système de refroidissement, l'opérateur se connecte à l'application comme il est indiqué dans la figure ci-dessous :

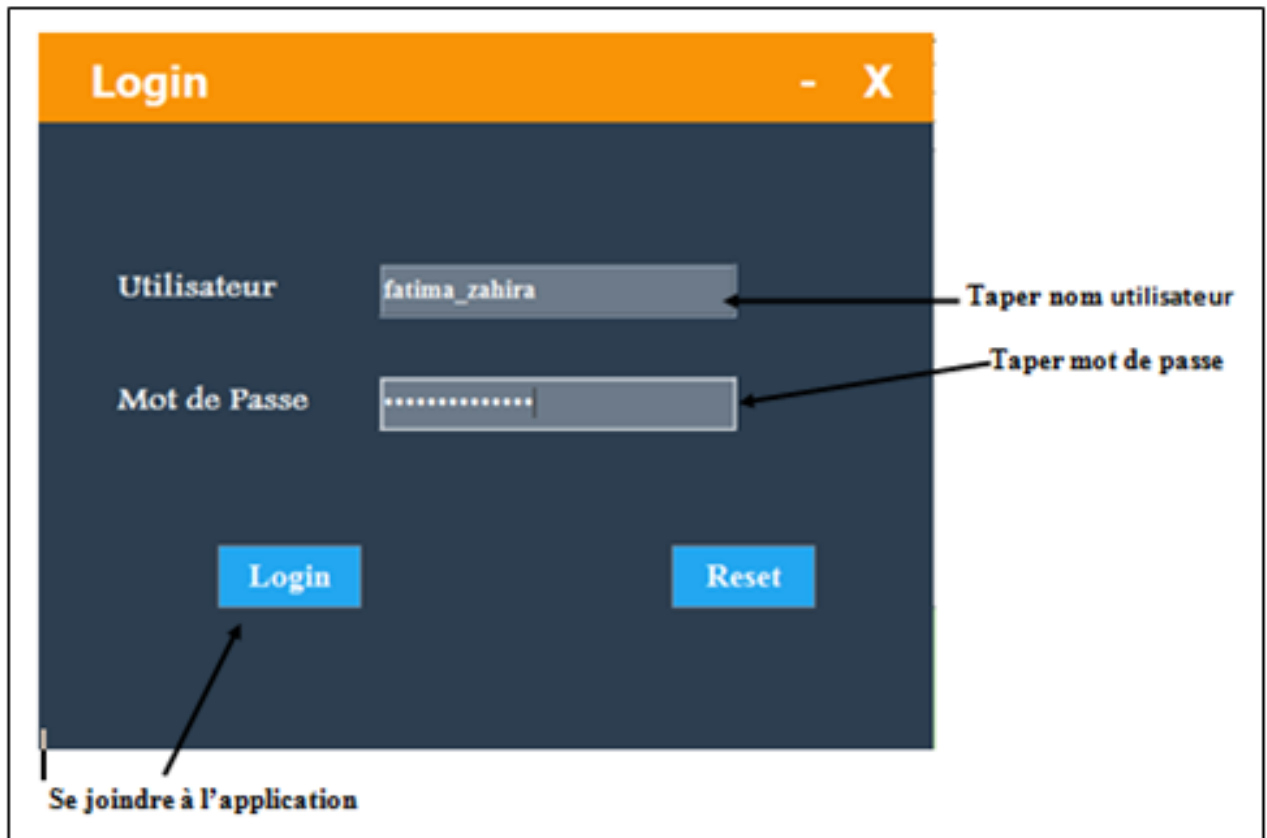


FIGURE 3.16 – Authentification de l'opérateur

La Figure 3.17 montre l'interface accueil et ses différentes fonctionnalités :

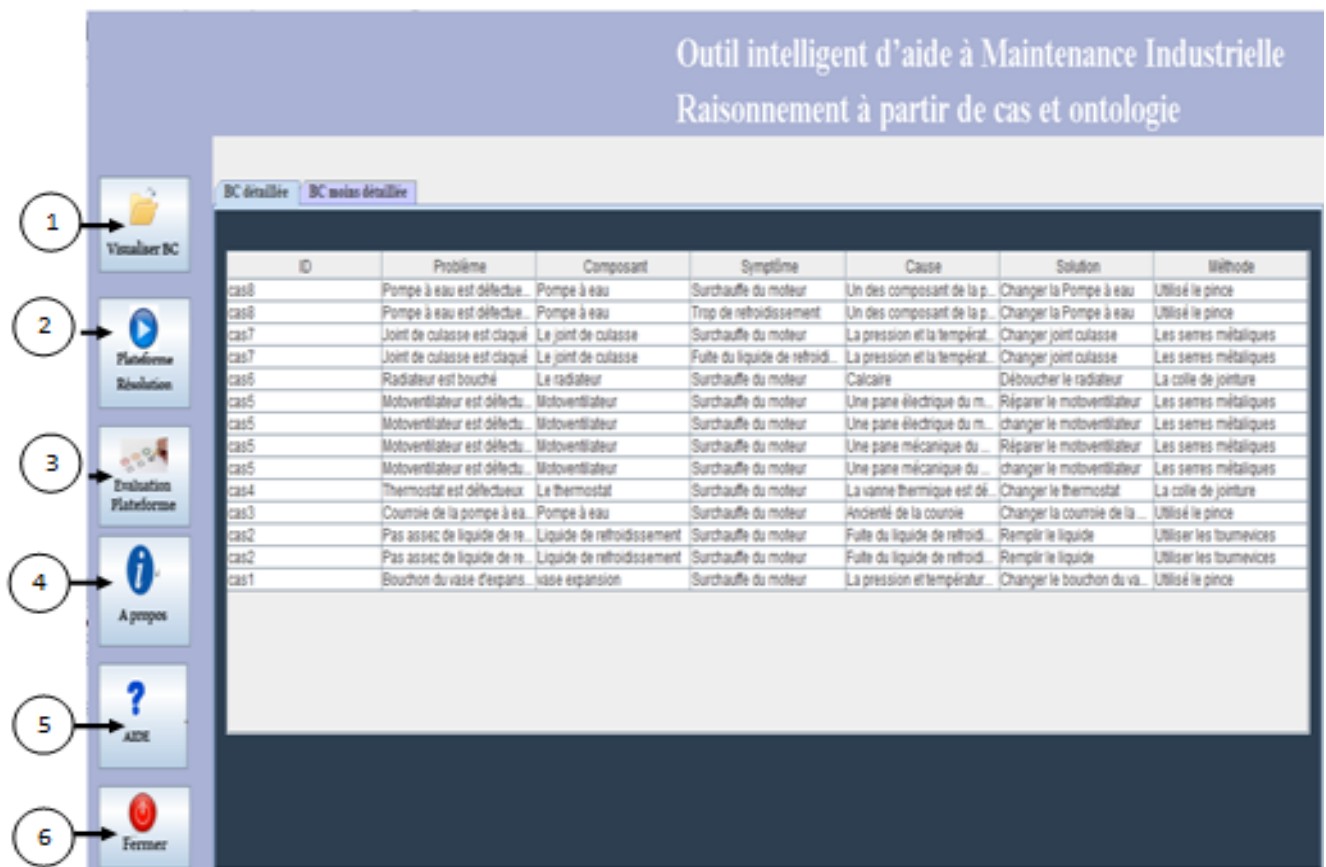


FIGURE 3.17 – Interface Accueil

Description ( flèches : 1 - 6 )

1. Ouvrir la base de connaissance ;
2. Lancer la plateforme résolution ;
3. Évaluation de la plateforme ;
4. A propos (créateurs de l'application) ;
5. Guide d'utilisation de notre application Onto-Master ;
6. Fermer l'interface accueil.

Dès que l'opérateur accède à l'espace de travail, il peut poster le pb. A partir de là le processus de résolution d'aide au diagnostic est en exécution.

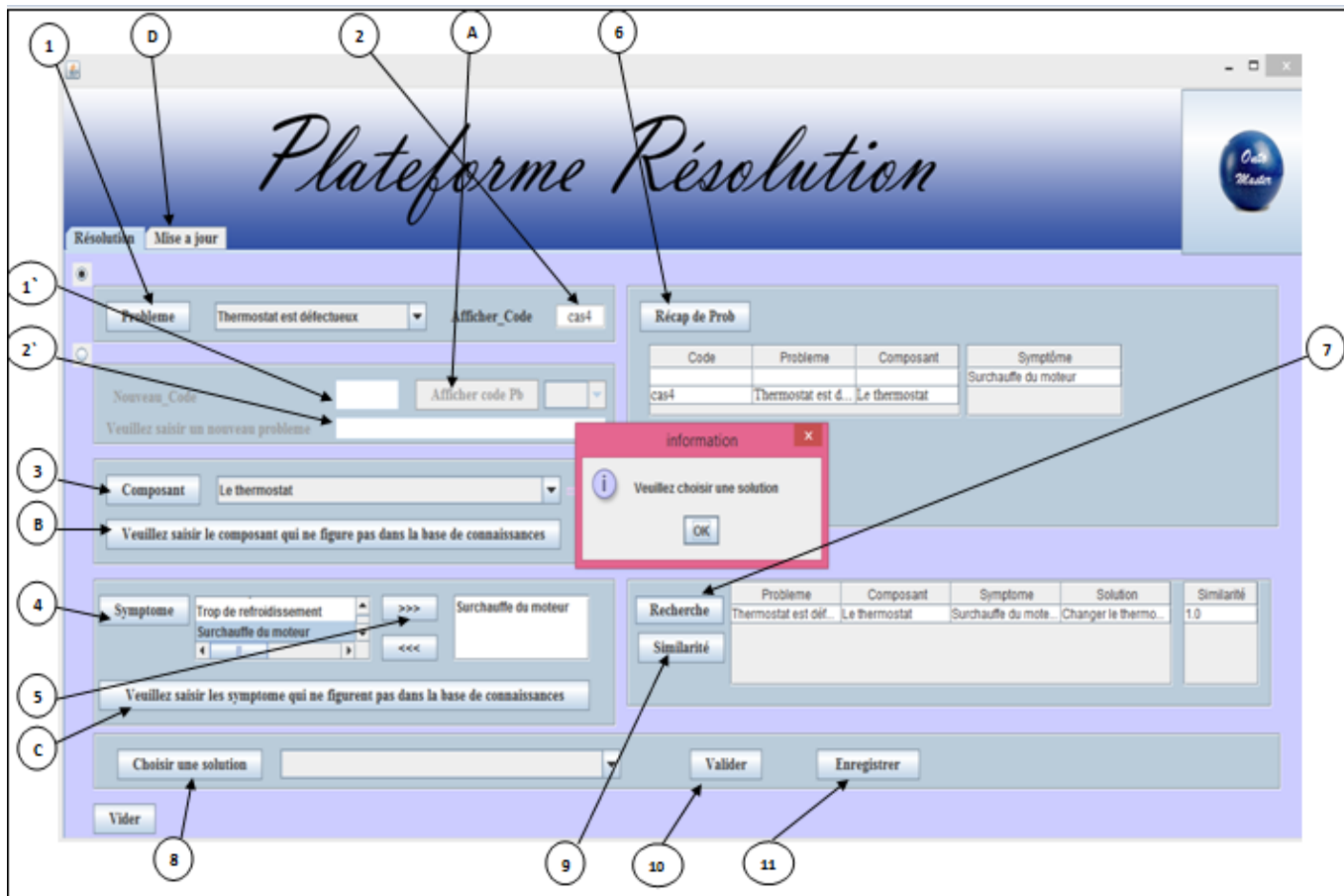


FIGURE 3.18 – Résolution d'un problème

**Etapas de résolution :**

**1<sup>er</sup> cas de figure (Pb existant) :**

1. Cliquer sur le bouton "Problème" pour afficher tous les problèmes existants dans la base.
2. Le code problème est affiché automatiquement.
3. Cliquer sur le bouton "Composant" pour afficher tous les composants qui existent dans la BC pour pouvoir en sélectionner un.
4. L'aperçu de tous les symptômes existants dans la BC se fait en cliquant sur le bouton "Symptôme" afin de sélectionner un ou plusieurs.
5. Choix du symptôme.
6. Pour afficher le récapitulatif du cas cible à savoir le Pb, composant et symptôme, on clique sur le bouton "Récap-Pb".



7. Pour afficher toutes les solutions possibles, on clique sur le bouton "Recherche-solutions" .
8. Choix de la/les solution(s) proposée(s).
9. Afin d'avoir plus d'exactitude de similarité de cas cible par rapport aux cas sources, on calcule le degré de similarité en appuyant sur le bouton "Similarité".
10. Validation de la solution choisie.
11. Enregistrement de la solution.

Dans le cas où 3 et/ou 4 ne sont pas vérifiés (symptôme et/ou composant n'existe pas dans la BC), on procède comme suit :

B. Mise à jour composant(voir détail dans la section de la page 63).

C. Mise à jour symptôme (voir détail dans la section de la page 63).

Dans le cas contraire à 7 (pas solutions similaires), on clique sur :

C. Mise à jour(voir page 62).

**2<sup>ème</sup> cas de figure(Pb inexistant) :**

A. Visualisation des codes des problèmes précédents enregistrés déjà dans la BC afin d'en créer un nouveau.

1'. Introduction du nouveau code-Pb.

2'. Saisie du Pb.

les étapes de résolution de 3 à D sont les mêmes.

**Détail de la mise à jour :**

Si on trouve pas un cas similaire alors on procède comme suit :

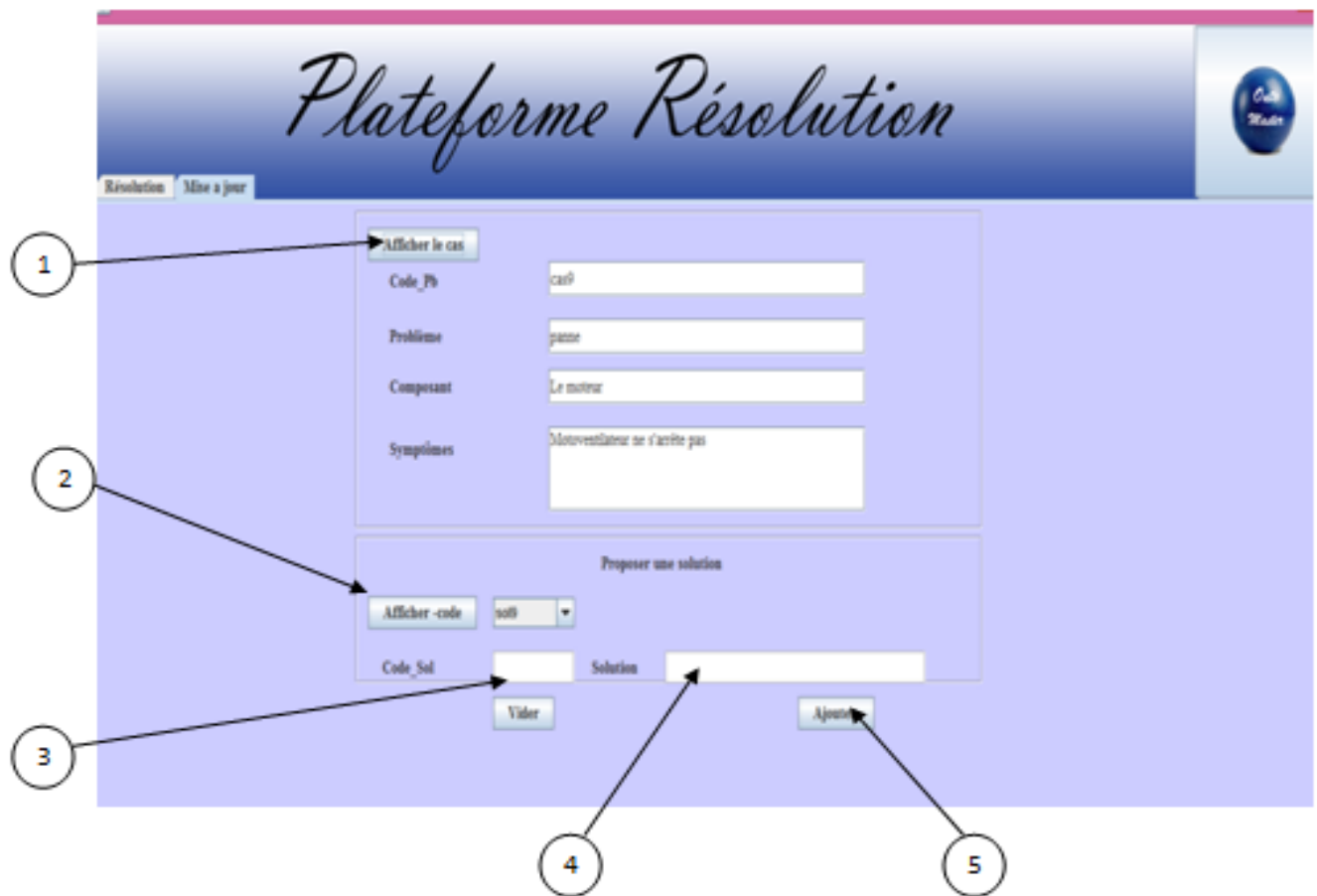


FIGURE 3.19 – Interface mise à jour

1. Ce bouton désigne l'affichage de cas (code-Pb, Problème, composant et symptôme)
2. Pour se rappeler du dernier code d'une solution enregistrée, on clique sur le bouton "Afficher-code" pour introduire un nouveau code solution.
3. Entrer un nouveau code solution.
4. Il s'agit de proposer une nouvelle solution.
5. Le bouton "Ajouter" sert à l'ajout du nouveau cas dans la BC(enrichissement).

Les figures ci-dessous (FIGURE 3.20 et FIGURE 3.21 )indiquent la mise à jour d'un composant ou symptôme .

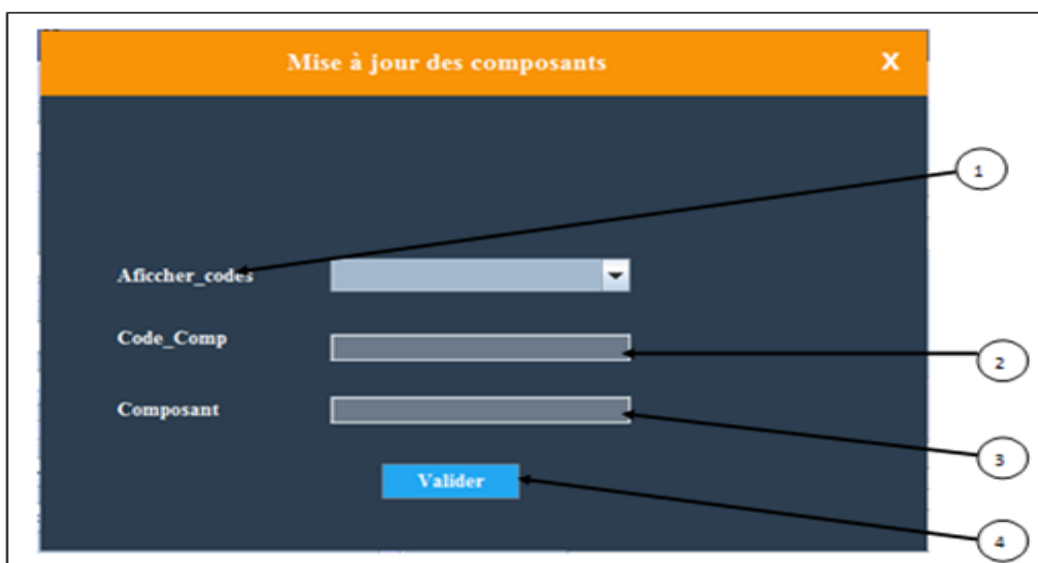


FIGURE 3.20 – Ajout d'un nouveau composant à la BC



FIGURE 3.21 – Ajout d'un nouveau symptôme à la BC

Récapitulons les fonctionnalités des deux interfaces ci-dessus :

1. De la même manière, on se rappelle des codes des composants ou symptômes existants déjà dans la BC, on clique sur le bouton "afficher code"
2. Saisie d'un nouveau code pour le composant ou symptôme.
3. Renseigner cette zone par un nouveau composant ou symptôme.
4. Finalement la validation de l'ajout se fait par le biais du bouton "Valider".

### 3.3.4 Exemple d'application

On considère la BC suivante :

**- Introduction du problème :**

Nous prenons comme exemple la description du problème suivante : "Pas assez de liquide de refroidissement", il est caractérisé par les symptômes suivants :

"symp1"

symp1 : Surchauffe du moteur.

**- Recherche(remémoration) :**

Dans cette phase, il s'agit de rechercher le(s) cas similaire(s) au problème posé.

On se basant sur l'exemple considéré, le cas recherché est relatif au cas2.

ID	Problème	Composant	Symptôme	Solution
Cas2	Pas assez de liquide de refroidissement	liquide de refroidissement	Surchauffe du moteur	Remplir le liquide

TABLE 3.1 – Cas similaire trouvé

**- Raisonnement**

Le problème similaire retrouvé à partir de la BC est proposé au décideur.

Puisque on note une similarité totale on a qu'à appliquer la solution.

A travers cet exemple d'application nous avons traité un seul cas de figure parmi plusieurs tels que : problème inexistant, partiellement défini, symptôme inexistant ou composant inexistant, qui seront le sujet des démonstrations pratiques ultérieures.

## 3.4 Conclusion

Ce chapitre rapporte en général les différentes étapes que nous avons suivies pour la conception et la mise en œuvre de notre application. Il définit les outils de développement tels que Protégé, Netbeans, etc. nous avons suivi la démarche de conception citée dans la partie I. Nous avons illustré par quelques pages écrans notre plateforme d'application et quelques scénarios d'exécution. Et nous avons terminé par un exemple d'application.

# Conclusion Générale

Le raisonnement à partir de cas, basé sur la réutilisation ou la modification des expériences passées est un fort paradigme et une approche habituellement utilisée dans le raisonnement humain pour la résolution de problèmes. Et donc, maintenir ces expériences antérieures dans une base de cas et les réutiliser permet de porter une vision sur l'intégration de cette méthodologie (RàPC) dans la capitalisation de connaissances et de savoirs faire des entreprises. Pour réaliser ces derniers objectifs, l'utilisation du (RàPC) dans notre projet, a prouvé que cette technique répond aux besoins scientifique et industriel.

Nous avons en conséquence étudié les fondements génériques de raisonnement à partir de cas. Dans la première étape, nous avons expliqué ses origines et son historique ainsi qu'étudier ses principes. De plus, nous avons précisé notre contexte de recherche en utilisant le raisonnement à partir de cas dans le cadre du diagnostic et de la résolution de panne à caractère industriel.

Nous avons proposé une méthodologie de démarche de capitalisation de connaissances pour la résolution des problèmes en renforçant le RàPC par les ontologies qui était très fructueux pour l'amélioration de la fiabilité de cette méthodologie. Par l'exploitation de ces premières (ontologies), la recherche des cas similaires en se basant sur les concepts a influencé, sûrement, la pertinence des résultats de cette recherche. Tous les principes, concepts, méthodologie, développement... etc relatifs aux ontologies ont été bien détaillés dans ce mémoire.

Par l'examinassions du cycle de RàPC, nous nous somme placées, sur la phase de l'élaboration et de la remémoration, qui ont été considérées comme des étapes clé dans notre processus du RàPC car ce sont les étapes où le problème se définit et que le rapprochement entre des cas sources (expérience passée) et une situation nouvelle

(le cas cible) se produit. Plus précisément, nous avons proposé une formalisation et une représentation d'un cas dans un système de diagnostic industriel. Dans la phase de l'élaboration, un nouveau cas est créé et la description du problème est ensuite complétée en collectant des informations pertinentes pour le problème à résoudre.

Pour ce faire, dans notre cas qui touche le diagnostic, nous avons spécifié l'ensemble des symptômes (descripteurs) adéquats afin d'identifier le problème à résoudre lors de la recherche de cas. Ces symptômes sont considérés comme pertinents car ils présentent des critères de recherche dans la base de cas. Pour modéliser les cas, nous avons utilisé les principes de formulation ontologique.

Le système que nous avons réalisé a le caractère d'une interactivité très forte. La phase de la remémoration a pour objectif, partant de l'ensemble de toutes les expériences passées, de préciser à chaque interaction avec l'opérateur (décideur/utilisateur..) le sous-ensemble de cas susceptibles de correspondre à la définition courante du problème de panne. Le cas le plus proche (similaire) est celui qui a besoin de moins d'effort d'adaptation.

Nous pouvons encore noter quelques difficultés dans la démarche de cette étude. Pour des équipements différents, cette approche se doit d'être adaptée. Ainsi, la collecte d'informations auprès des techniciens du domaine étudié peut s'avérer parfois problématique.

Parallèlement, nous pouvons projeter quelques axes de recherche :

- Étendre la plateforme de sorte que le travail soit réparti en groupe.
- Il serait intéressant d'appliquer d'autres phases du cycle du RàPC telles que la révision et maintenance du système.
- Il serait même possible de prendre en compte d'autres critères lors de la phase de remémoration, tels que les causes qui ont déclenché un problème donné.
- De même nous pouvons envisager des méthodes pour l'application de la (s) solution (s).
- On peut également appliquer ces travaux dans un autre contexte or que le diagnostic industriel comme le secteur des services publics.

## CONCLUSION GENERALE

---

Enfin, nous souhaitons que notre réflexion puisse contribuer à ouvrir d'amples pistes de recherche et à orienter leurs travaux pour épanouir notre domaine de l'informatique et plus précisément nos collègues étudiants dans notre formation en master Informatique de l'option « Réseaux et Ingénierie des Données »



# Bibliographie

- [AAMO 94] Aamodt A. et Plaza E. "Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches". AI Communications. IOS Press, Vol. 7 : 1, pp. 39-59, 1994.
- [ABDE 07] Abdel Kader KEITA. "Conception Coopérative d'Ontologies Pré-Consensuelles : Application au domaine de l'Urbanisme". Thèse de doctorat, l'Institut National des Sciences Appliquées de Lyon, 2007.
- [AFNO 01] Afnor. "Maintenance terminology". Norme française homologuée, NF EN 13306, 2001.
- [AHME 18] Amedeo. Napoli. "Système de Recommandation de Cours à Base d'Ontologie". These de Doctorat, université de Tlemcen (Algérie), 2018.
- [AMED 97] Amedeo. Napoli. "Une introduction aux logiques de descriptions". Rapport de recherche. Institut national de recherche en informatique et en automatique, 1997
- [BACH 00] Bachimont, B. "Engagement sémantique et engagement ontologique : conception et réalisation d'ontologie en ingénierie des connaissances". Direction de la Recherche Prospective Institut National de l'Audiovisuel 4, Avenue de l'Europe 94366, 2000.
- [BAND 10] Melle BANDOUC FARIDA. "Diagnostic de pannes d'un système de pompe photovoltaïque". Mémoire de magister, université Mouloud Mammeri de Tizi-Ouzou, 2010.
- [BAND 11] Melle BANDOUC FARIDA, A. Hadj Arab, K. Bouchouicha et N. Zerhouni. "Division Energie Photovoltaïque". Revue des Energies Renouvelables Vol. 14 N 3 (2011)p.543 – 560.
- [BANE 07] BANEYX Audrey, CHARLET Jean. "Évaluation, évolution et Maintenance d'une ontologie en médecine : état des lieux et expérimentation", université paris, 2007.

- [BELA 15] BELABBES FATIMA. "Conception et implémentation d'une ontologie médicale Cas : insuffisance cardiaque". Mémoire de master, université Abou Bakr Belkaid– Tlemcen, 2015.
- [BELLA 99] Belacel. "Méthode de classification multicritère : Méthodologie et applications à l'aide au diagnostic médical". Thèse de doctorat, université libre de Bruxelles, 1999.
- [BENA 17] Mr BENABDALLAH Ali. "Construction semi-automatique des ontologies à partir des documents textuels arabes". Thèse de doctorat, université abou-bekr belkaid – Tlemcen, 2017.
- [BENB 12] BENBELKACEM Sofia. "Aide à la décision médicale : Planification guidée par Datamining et Raisonnement à Partir de Cas". Mémoire de magister, université d'Oran, 2012.
- [BOUA 16] [SOUA16] BOUAICHA Souad. "Un nouveau formalisme pour la représentation et le raisonnement sur une ontologie hybride". Thèse de doctorat, université Constantine2 Abdelhamid Mehri, 2016.
- [CLER 01] Cleret M., Le Beux P., Le Duff F. "Les systèmes d'aide à la décision médicale". Les Cahiers du numérique, Vol. 2, pages 125 à 154, 2001.
- [DEND 14] Dendani-Hadiby Nadjette. "Capitalisation et réutilisation des connaissances de diagnostic de panne des turbines". Thèse de doctorat, université Badji Mokhtar- Annaba 2014.
- [DUBU 01] Dubuisson B., Boutleux E., Dague P., Denoeux T., Didelet E., Gandvalet Y. et Masson M. "Diagnostic, intelligence artificielle et reconnaissance de formes". Éditeur(s) : Hermès. Ouvrage, nb. de pages : 286, année 2001.
- [FERN 97] FERNANDEZ M., A. GOMEZ-PEREZ et al. "METHONTOLOGY : from ontological art owards ontological engineering". Conférence : AAAI-97 Spring Symposium Series, 24-26 March 1997. Stanford University.
- [FUCH 99] Fuchs B., Lieber J., Mille A., et Napoli A. "Towards a unified theory of adaptation in case-based reasoning". In Case Based Reasoning Research and Development. International Conference on Case-Based Reasoning (ICCBR 1999) : Case-Based Reasoning Research and Development pp 104-117, 1999.
- [GHER 17] GHERZOULI Imane. "Système d'aide au diagnostic médical à base d'ontologie". Mémoire de Master, université Mohamed Boudiaf - M'sila 2017.
- [GOME 99] A. Gómez-Pérez. "Ontological engineering : A state of the art". Expert Update : Knowledge Based Systems and Applied Artificial Intelligence", v. 2 (n. 3) ; pp. 33-43, 1999.

- [GUAR 95] Guarino N., et Giaretta P. "Ontologies and Knowledge Bases : Towards a Terminological Clarification". Slightly amended version of a paper appeared in N.J.I. Mars (ed.), Towards Very Large Knowledge Bases, IOS Press 1995, Amsterdam.
- [HACI 14 ] HACINE GHERBI Ahcine. "Construction d'une Ontologie pour le WEB Sémantique". Mémoire de magister, université Ferhat Abbes –Sétif1- 2014.
- [HADJ 12] HADJOUI Fatima Zohra. "Gestion des Ontologies Médicales". Mémoire de master, université Abou Bakr Belkaid– Tlemcen, 2012.
- [HAST 01] Hastie, Tibshirani, Friefman. "The Elements of Statistical Learning : Data Mining, Inference and Prediction".Second Edition. Ouvrage, 2001.
- [ISSA 05] Isaac. "Conception et utilisation d'ontologies pour l'indexation de documents audiovisuels". Thèse de doctorat, université Paris IV, 2005.
- [IVAN 06] 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 doctorat, université de Franche-Comté, 2006.
- [KHAL 09] [SOUH 09] Souheila KHALFI. "Construction d'une ontologie pour la prise en charge des patients à domicile". Mémoire de Magister, université Mentouri Constantine, 2009.
- [LIEB 04] Lieber J, d'Aquin M, Brachais S et Napoli A. "Une étude comparative de quelques travaux sur l'acquisition des connaissances d'adaptation pour le raisonnement à partir de cas". 12ème Atelier de Raisonnement à Partir de Cas (RàPC'04), pages 53–60, Villetaneuse, France, Mars 2004.
- [LOPE 05] Lopez de Mantaras R, McSherry D, Bridge D, Leake D, Smyth B, Craw S, Faltings B, Maher M.L, Cox M, Forbus K, Keane M, Aamodt A et Watson I. "Retrieval, Reuse, Revise, and Retention in CBR". Knowledge Engineering Review. Volume 20, Issue 3 September 2005 , pp. 215-240, 2005.
- [LORA 08] Loraine MARCHEIX. "Conception d'une ontologie à partir d'un thésaurus spécialisé dans le domaine de l'archéologie et des sciences de l'Antiquité". Mémoire de Master, université Vincennes – Saint-Denis, Paris 8, 2008.
- [MACA 02 ] Macaire Ahlonsou, Emmanuel Blanchard Henri Briand, Fabrice Guillet. "Transformation des concepts du diagramme de classe UML en OWL full". RNTI-E-5, LINA – Université de Nantes, 2002.
- [MARI 99] Mariano Fernandez Lopez. "Building a Chemical Ontology Using Methontology and the Ontology Design Environment". Journal, 1999.

- [MEKR 09] MEKROUD Nouredine. "Intégration des techniques du Datamining et des bases de données avancées dans le processus de Gestion des Connaissances : proposition d'un processus hybride basé sur le raisonnement à partir de cas". Mémoire de Magister, université Ferhat Abbes -Sétif- 2009.
- [MILL 99] Mill. A. "Tutorial CBR : Etat de l'art de raisonnement à partir de cas". Plateforme AFIA'99, Palaiseau, 1999.
- [MILL 06] Mille A. "Traces based reasoning (TBR) definition, illustration and echoes with story telling". Rapport Technique RR-LIRIS-2006-002, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon, 2006.
- [MILL 96] Mille A., Fuchs B. et Herbeaux O. "A unifying framework for Adaptation in Case-Based Reasoning". Conférence : Proceedings of the ECAI'96 Workshop : Adaptation in Case-Based Reasoning, p. 22-28, 1996.
- [MOHA 09] Mohamed Karim Haouchine. "Remémoration guidée par l'adaptation et maintenance des systèmes de diagnostic industriel par l'approche du raisonnement à partir de cas". Thèse de doctorat, université de Franche-Comté 2009.
- [NECH 91] Neches R, Fikes R. E, Finin T, Gruber T. R, Senator T. et Swartout W. R. "Enabling technology for knowledge sharing". Journal : American Association for Artificial Intelligence Menlo Park, volume 12 Issue 3, Fall 1991, pages 36-56.
- [NEGR 91] Negrello. "Systèmes experts et intelligence artificielle". Cahier Technique : Merlin Gerin nA 157 / p.2, université Paul Sabatier de Toulouse, 1991.
- [NOUR 17] Nourhène Ben Rabah, Ramla Saddem, Faten Ben Hmida, Véronique Carre-Menetrier<sup>1</sup> et Moncef Tagina. "Approche originale utilisant le Raisonnement à Partir de Cas pour le Diagnostic en Ligne des Systèmes Automatisés de Production". 11ème Colloque sur la Modélisation des Systèmes Réactifs, Marseille, France, 2017.
- [PSYC 03] Psyché V, Mendes O. et Bourdeau J. "Apport de l'ingénierie ontologique aux environnements de formation à distance". Rapport de recherche, volume 10 2003.
- [PIEC 03] Piechowiak S. "Intelligence artificielle et diagnostic. Techniques de l'ingénieur, S 7 217, décembre, 2003.
- [SCHM 89] Schmidt-Schauß and Smolka, Gert Smolka. "Attributive concept descriptions with unions and complements". Book, volume 68, 1988.

- [STAN 09] Stan,S.d. "Génétic algorithms for werks -pace optimization of planar medical parallel robot". Conference :Soft Computing and Pattern Recognition ,Page(s) : 155 - 160, 2009.
- [STUD 98] Studer R, Benjamins V. R. et Fensel. "Knowledge engineering : Principles and Methods. Data Knowledge Engineering", volume 25, Issues 1–2, March 1998, Pages 161-197, 1998.
- [SWAR 97] Swartout B, Patil R, Knight K. et Russ T. "Towards Distributed Use of Large Scale Ontologies". Rapport technique :Ontological Engineering. AAAI-97 Spring Symposium Series, page 138-148. (1997)
- [USCH 96] Uschold M et Grüninger M. "Ontologies : Principles, Methods and Applications". Rapport technique :To appear in Knowledge Engineering Review ,volume 11 Number 2 ,1996.
- [WELT 01] Welty C.,Guarino N. "Supporting ontological analysis of taxonomic relationships". Data and Knowledge Engineering, volume 39, Issue 1, October 2001, Pages 51-74.
- [WILK 98] Wilke W. et Bergmann R. "Techniques and Knowledge Used for Adaptation During Case-Based Problem Solving". 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 1998.

