

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Chadli Bendjedid

Faculté des Sciences et de la Technologie

Département d'Informatique



الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة الشاذلي بن جديد

كلية العلوم والتكنولوجيا

قسم الآلة

MEMOIRE

Présentée par

MAHMOUDI ROUFAIDA

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

**Réalisation d'un Système Expert d'aide au diagnostic de pannes d'un
Micro-ordinateur**

Soutenue le :

Devant le Jury composé de :

Nom et Prénom

Grade

Mme. FERROUM AssiaMCB

Université Chadli Bendjedid El-Tarf

Président

Mme. BOUGUERNE ImenMCB

Université Chadli Bendjedid El-Tarf

Rapporteur

Mme. MAKHLOUF AminaMCB

Université Chadli Bendjedid El-Tarf

Examineur

Année Universitaire : 2019/2020

Remerciements

Merci Allah (mon dieu) de nous avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever nos mains vers le ciel et de dire « Ya Allah ».

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement du Dr. Bougeurne Imen, je le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant ma préparation de ce mémoire.

Mon remerciement s'adresse aussi aux membres du jury qui ont accepté d'examiner mon travail.

A tous mes enseignantes, enseignants et la composante de la ressource humaine du département informatique de l'Université Chadli Bendjedid.

Enfin et avec un très grand plaisir, je remercie tous mes proches et ami(e)s, qui j'ai toujours soutenu et encouragé au cours de l'élaboration de ce mémoire.

Avec plaisir, je remercie tous celles et ceux qui ont participé de près ou de loin à la réalisation de ce travail.

Merci à toutes et à tous.

Avant tout je rends grâce à dieu de m'avoir donné la force et le courage d'achever ce travail, je dédie ce modeste travail à :

Mes très chers parents Mon père «Mahmoudi Mabrok» et ma mère «Djelabe Akila » pour leur encouragements et leurs soutiens qui sont fière de trouve cette résultat de longues année de sacrifices. Aucune dédicace ne saurait exprimer l'amour, l'estime et le respect que j'ai toujours eu pour vous.

A mes très chers frères «Borhane Eddine» et «Mosaabe» et «Souhaibe» je te souhaite une vie prospère pleine d'amour et de joie, que la vie ne puisse jamais nous séparer.

A toute les familles

A mon encadreur : « Mme. Bougeurne Imene » pour son soutien au moment difficile de mon travail.

Mahmoudi Roufaida

Table des matières

Remerciements	2
Dédicace	3
Table des matières	4
Tables des figures	7
Liste des tableaux	9
Liste des acronymes	10
Introduction Générale.....	11
1. problématique.....	11
2. Objectif de mémoire.....	11
3. Contenu de mémoire	11
Chapitre 1 : Etat de l' Art	13
1. Introduction	13
2. L'intelligence Artificielle.....	13
2.1. Définition de l'IA	13
2.2. Disciplines et Domaines d'application de l'IA.....	14
2.3. Les avantages et les inconvénients de l'IA.....	15
3. Système expert.....	15
3.1. Définition de SE	15
3.2. Les caractéristiques du système expert.....	16
3.3. Architecture générale d'un système expert	17
3.4. Les domaines d'application d'un SE	20
3.5. Quelques exemples et évolution de SE.....	21
3.6. Les avantages de système expert	21
3.7. Les inconvénients de système expert.....	21
4. Représentation des connaissances.....	22
4.1. Introduction	22

4.2.	Type de représentation des connaissances.....	22
5.	Structure Caractéristique d'un moteur d'inférence.....	30
5.1.	Le cycle de travail de moteur d'inférence	30
5.2.	Mode de résonnement.....	31
6.	Stratégie de recherche	34
6.1.	La Recherche en profondeur d'abord	34
6.2.	La Recherche en largeur d'abord.....	34
6.3.	La Recherche en profondeur limité	34
6.4.	La Recherche heuristique	34
7.	Stratégie de contrôle	35
7.1.	Régime de contrôle.....	35
7.2.	Critère de monotonie	35
8.	Conclusion	35
	Chapitre 2 : conception de l'application	36
1.	Introduction	36
2.	Architecture du système	36
2.1.	Description générale du système SEDIP :.....	36
2.2.	Description détaillée du système SEDIP	37
A.	L'étape d'acquisition des connaissances :.....	37
B.	L'étape d'effectuer la recherche :.....	38
C.	L'étape de traitement et raisonnement :.....	38
D.	L'étape de communication de solution:.....	38
3.	Conception UML.....	39
3.1.	Introduction	39
3.2.	Les diagrammes de notre système	39
A.	Diagramme de cas d'utilisation :.....	39
B.	Diagramme de séquence :.....	44
C.	Diagramme de cas classe :.....	55
4.	Comparaison entre les travaux existant.....	57
5.	Conclusion.....	58

Chapitre 3 : Réalisation et Test	59
1. Introduction	59
2. les outils et les langages utilisés	59
2.1. Les outils utilisés	59
A. Eclipse (l'environnement de développement)	59
2.2. Les langage utilisés.....	61
3. L'organigramme de notre application	62
4. Présentation et test de l'application.....	62
4.1. Page d'authentification	63
4.2. Menu principale.....	63
4.2.1. Espace expert	63
4.2.1.1. Menu de gestion de la base des faits	64
4.2.1.2. Menu de gestion de la base des règles.....	66
4.2.1.3. Menu de gestion des recherches et inférences	67
4.2.2. Espace utilisateur	68
4.2.2.1. Menu de gestion des recherches et inférences	69
4.2.2.2. Consulter le fichier help	70
4.3. A propos de SEDIP :	70
5. Conclusion.....	71
Conclusion et perspectives	72
Références	73
A. Références Bibliographiques.....	73
B. Références Web (Techniques).....	74

Tables des figures

Figure 1.1 : Acteurs principaux de système expert	16
Figure 1.2 : Architecture d'un Système expert.	18
Figure 1.3 : Classification de J.L. Laurière.	23
Figure 1.4 : Représentation d'un réseau sémantique.....	26
Figure 1.5 : Représentation d'un réseau de neurones.....	27
Figure 1.6 : Représentation de la classe candidat.....	28
Figure 1.7 : Exemple d'héritage.	29
Figure 1.8 : Exemple de polymorphisme.	29
Figure 1.9 : Représentation de cycle de travail de moteur d'inférence.....	31
Figure 1.10 : Représentation du résultat de l'exemple.....	33
Figure 2.1 : L'architecture de notre système.....	37
Figure 2.2 : Représentation de stockage des données.....	38
Figure 2.3 : Diagramme de cas d'utilisation Expert _ Utilisateur.....	41
Figure 2.4 : Diagramme de séquence de cas d'utilisation (Authentification)..	45
Figure 2.5 : Diagramme de séquence de cas d'utilisation (gestion de base de faits 1)..	46
Figure 2.6 : Diagramme de séquence de cas d'utilisation (gestion de base de faits 2)..	47
Figure 2.7 : Diagramme de séquence de cas d'utilisation (gestion de base de règles 1).. ...	48
Figure 2.8 : Diagramme de séquence de cas d'utilisation (gestion de base de règles 2)	49
Figure 2.9 : Diagramme de séquence de cas d'utilisation (gestion de système expert)..	50
Figure 2.10 : Diagramme de séquences de cas d'utilisation-Expert (Faire des recherches et inférences).....	51
Figure 2.11 : Diagramme de séquences de cas d'utilisation-Expert (Charger un projet)....	52
Figure 2.12 : Diagramme de séquences de cas d'utilisation (Sauvegarder un projet).....	53
Figure 2.13 : Diagramme de séquences de cas d'utilisation-Utilisateur (Charger un projet).	54
Figure 2.14 : Diagramme de séquences de cas d'utilisation-Utilisateur (Faire des recherches et inférences).....	55
Figure 2.15 : Diagramme de classe.	57
Figure 3.1 : Logo Eclipse..	59
Figure 3.2 : Eclipse IDE.....	60
Figure 3.3 : Logo java JDK.....	60
Figure 3.4 : Logo StarUML.....	61

Figure 3.5 : Logo Pcestar.....	61
Figure 3.6 : Logo Java.....	62
Figure 3.7 : Organigramme de l'application.....	62
Figure 3.8 : Page d'authentification.....	63
Figure 3.9 : Espace Expert.....	64
Figure 3.10 : Gestion des faits (vide).....	65
Figure 3.11 : Gestion des faits (rempli).....	65
Figure 3.12 : Gestion des règles (vide).....	66
Figure 3.13 : Gestion des règles (rempli).....	66
Figure 3.14 : Gestion de recherche.....	67
Figure 3.15 : Gestion de recherche avec chainage avant.....	67
Figure 3.16 : Gestion de recherche avec chainage arrière.....	68
Figure 3.17 : Espace utilisateur.....	68
Figure 3.18 : Gestion de recherche avec chainage avant.....	69
Figure 3.19 : Gestion de recherche avec chainage arrière.....	69
Figure 3.20 :Fichier helpe	70
Figure 3.21 : A propos de SEDIP	71

Liste des tableaux

Table1.1 : Quelques caractéristiques du système expert	18
Table1.2 : Quelques domaines d'application d'un système expert.....	21
Table2.1 : Description détaillé pour les cas d'utilisation coté Expert.....	44
Table2.2 : Description détaillé pour les cas d'utilisation coté Utilisateur..	44
Table2.3 : Comparaison entre les systèmes experts..	59

Liste des acronymes

IA	Intelligence Artificiel
SE	Système Expert
TALN	Traitement Automatique du Langage Naturel
MI	Moteur Inférence
SEDIP	Système Expert Diagnostic Pannes
UML	Unified Modeling Language ; <i>Langage de Modélisation Unifié</i>
BDF	Base De Faits
BDR	Base De Règles
OOSE	Object Oriented Software Engineering

Introduction Générale

Depuis la découverte des ordinateurs, plusieurs chercheurs ont essayé de donner une notion d'intelligence aux programmes informatiques. Cela mène à l'émergence d'un nouveau concept appelé intelligence artificielle(IA). Au début, plusieurs critiques ont été posées concernant l'intelligence artificielle, ensuite et avec le temps, l'IA s'est investie dans nombreux domaines où l'informatique classique n'est pas applicable. Parmi lesquelles « le système expert ». Où le but de ce dernier était est de modaliser la démarche et le cheminement de la pensée d'un expert.

Aujourd'hui, l'être humaine est devenu fortement dépendant de l'ordinateur pour maintenir sa vie professionnelle et personnelle en y stockant des informations, mais parfois ce dernier est soumis à des pannes et c'est ce qui nous a poussés à réaliser un système expert qui nous aide à diagnostiquer les défaillances et faire le traitement pour résoudre les problèmes de ce genre.

1. problématique

Le problème que nous avons traité dans notre étude liées a la comment représenté le savoir de l'expert au domaine de la maintenance de l'ordinateur ?

A ce que la méthode intelligent de système expert répond a ce problème et faire le traitement ? Nous le découvrirons dans le reste des chapitres.

2. Objectif de mémoire

Notre objectif est de faire et réaliser un système intelligent (SE) capable de faire des taches d'analyse et pour identifier les menaces et les risques et détecté les pannes soit matériels ou logicielles. Le but de ce système est de réduire les erreurs humaines, il détecte les défaillances en peu de temps et nous donne un résultat précis et juste, de minimiser le déplacement vers les informaticiens au domaine.

3. Contenu de mémoire

Ce manuscrit est organisé comme suit : une introduction générale, trois chapitres, et une conclusion avec des perspectives.

Chapitre 1 : dans ce chapitre, nous réalisons une description détaillée sur l'intelligence artificielle et les systèmes expert.

Chapitre 2: est destiné à la conception et la modélisation UML et l'architecture de système.

Chapitre 3 : montrera la réalisation de notre application et les outils utilisés. Enfin, dans la conclusion nous donnerons le bilan de tout ce que nous avons appris lors du développement du système depuis la conception jusqu'à la réalisation.

1. Introduction

Depuis longtemps déjà, il est convenu de considérer deux approches de l'informatique : l'une dite algorithmique ou impérative et l'autre que l'on appelle l'Intelligence Artificielle (I.A.)

Les systèmes experts représentent une des applications prometteuses de l'intelligence artificielle. Ils sont utilisés dans plusieurs domaines d'activité humaines, par exemple : l'industrie, la médecine, l'agriculture, et le finance, où ils sont implémentés pour remplacer l'expert humain, pour être constituer un support de travail pour les utilisateurs du domaine et pourquoi pas un bon moyen pour les amateurs de devenir eux-mêmes des experts [1].

Dans ce chapitre nous allons en premier, donner une bref définition sur les deux concepts l'IA et SE et leurs domaines d'applications et quelques avantages et inconvénients, Nous citerons ensuite les éléments de base de l'architecture des SE, nous enchainons par la suite par la description des différentes représentations des connaissances, Enfin nous terminons avec les étapes à suivre pour cycle de base d'un moteur d'inférence avec les différents stratégies de recherche.

2. L'intelligence Artificielle

2.1. Définition de l'IA

L'intelligence artificielle est un ensemble des méthodes, des outils, et des systèmes définis pour résoudre des problèmes dont leur solution nécessite de l'intelligence humaine [2].

Ainsi, on peut définir l'IA comme la solution informatique à utiliser quand l'algorithmique classique ne suffit pas pour résoudre un problème. En fait, l'intelligence artificielle est une discipline dont le but principal est de construire des systèmes dits intelligents, à savoir ceux qui sont capables de comprendre ou tout au moins de modéliser une situation donnée. Un système de ce type doit être capable de s'adapter à une situation, de faire des analogies et des projections d'idées. Tout cela requiert de souplesse et facilité au niveau de la description des objets ou des événements et des différentes stratégies à appliquer [3].

2.2. Disciplines et Domaines d'application de l'IA

Le sujet de l'IA couvre plusieurs domaines, Parmi ces domaines nous citons : [W1]

❖ Les systèmes experts

Il existe quelques applications qui intègrent la machine, la programmation et des données spéciales pour faciliter la réflexion. Ils donnent des éclaircissements et des exhortations aux utilisateurs.

❖ Les jeux

Elle joue un rôle important dans les jeux stratégiques que l'on parle des échecs ou de jeux vidéo. La machine devrait alors être en mesure de penser à plusieurs étapes possibles basées sur la connaissance heuristique.

❖ Les robots intelligents

Les robots sont l'une des meilleures créations humaines. Ils peuvent effectuer plusieurs tâches en un rien de temps. Bien qu'ils ne puissent pas remplacer les humains, ils sont très efficaces pour toutes les tâches répétitives et simples.

❖ La reconnaissance

Qu'elle que soit de la parole, de l'écriture ou du visage, l'IA a fait de nombreux progrès dans ces actions.

❖ L'apprentissage automatique

L'apprentissage automatique est un sous- domaine de l'intelligence artificielle (IA) qui se concentre sur la conception de système qui apprennent ou améliorent le rendement en fonction des données qu'ils consomment.

❖ Le traitement automatique du langage nature

On regroupe sous le vocable de traitement automatique du langage naturel (TALN) l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication [4].

🚦 L'IA est en interaction avec plusieurs sciences qui ont contribué à son développement, parmi elles, l'informatique, la psychologie, les sciences cognitives, la philosophie, la mathématique, l'économie, la médecine, l'industrie et l'ingénierie.

2.3. Les avantages et les inconvénients de l'IA

➤ Parmi les avantages de l'IA, nous avons : [W2]

- **La réduction des erreurs** : L'intelligence artificielle nous aide à réduire l'erreur humaine et les chances d'atteindre la précision avec un degré de précision supérieur. Il est appliqué dans divers domaines tels que l'exploration de l'espace.
- **Les travaux répétitifs** : Les tâches répétitives de nature monotone peuvent être effectuées à l'aide de l'intelligence artificielle. Les machines pensent plus vite que les humains et peuvent être soumises à plusieurs tâches.
- **L'exploration difficile** : L'intelligence artificielle et la science robotique peuvent être utilisées dans les processus miniers et autres processus d'exploration de combustibles. De plus, ces machines complexes peuvent être utilisées pour explorer le fond des océans et ainsi surmonter les limites humaines.

➤ Parmi les inconvénients de l'IA, nous avons :

- **Un coût élevé** : La création d'une intelligence artificielle nécessite des coûts énormes car ce sont des machines très complexes. Leur réparation et leur entretien impliquent des coûts importants également.
- **Le chômage** : Le remplacement d'êtres humains par des machines peut entraîner un chômage important.
- **Pas de créativité** : Ce ne sont pas la force de l'intelligence artificielle. Bien qu'ils puissent vous aider à concevoir et à créer, ils ne sont pas à la hauteur du pouvoir de penser que possède le cerveau humain ni même de l'originalité d'un esprit créatif.

3. Système expert

3.1. Définition de SE

D'une manière générale, un système expert est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine particulier. Il s'agit de l'une des voies tentant d'aboutir à l'intelligence artificielle.

En outre, Un système expert est un système d'aide à la décision basé sur un moteur d'inférence et sur une base de connaissances. Il est la transcription logicielle de la réflexion d'un expert dans un domaine donnée.

Plus précisément, un système expert est un logiciel capable de répondre à des questions, en effectuant un raisonnement à partir de faits et de règles connus. Il peut servir notamment comme outil d'aide à la décision. [5]

➤ Exemple

Les systèmes experts sont des systèmes basés sur les connaissances d'un expert humain. Par exemple un système expert de diagnostic des pannes d'un micro ordinateur (notre sujet de mémoire) possède une base de connaissances contenant des règles pour le diagnostic et la détection des pannes de la même manière du raisonnement d'un informaticien (réparateur). Alors, le système expert est un programme qui peut fournir une expertise pour la résolution d'un problème défini dans le même domaine de l'expertise originale [2].

- Un système expert est caractérisé par trois acteurs principaux, chacun jouant un rôle l'expert de domaine et les utilisateurs et le cogniticien (qui implémente les connaissances de l'Expert humain dans un langage informatique sous forme de règles). L'expert transfère son expérience au système expert afin que les utilisateurs bénéficient de cette expertise en utilisant le système expert.

La [Figure1.1] montre ça.

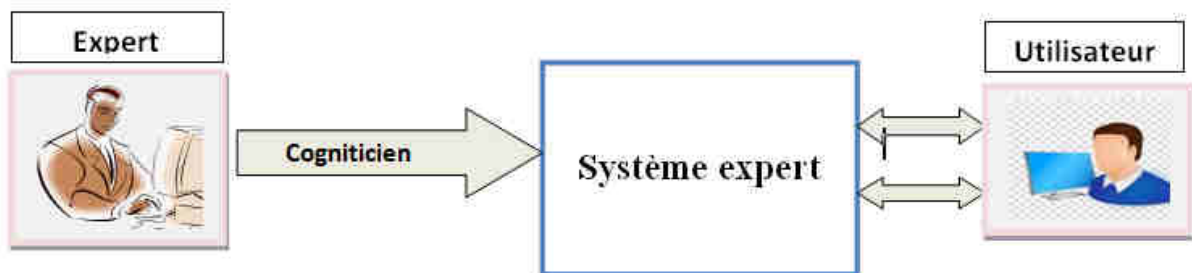


Figure1.1 : Acteurs principaux de système expert.

3.2. Les caractéristiques du système expert

Les systèmes experts est conçu pour supporter l'accumulation des connaissances d'un expert humain et de résoudre des problèmes complexes, Comme ce dernier se distingue par un ensemble des caractéristiques [W3, W4], nous les mentionnons dans le tableau suivante: [Tableau1.1]

Propriétés	Descriptions
Compétent	Le système a la connaissance pour répondre comme un VRAI Expert du domaine. Il peut aussi développer plus de connaissance que son Expert, mais il doit en garantir la véracité de sa connaissance.
Robuste	La réponse doit toujours être juste (conforme à l'Expert du domaine) et ne peut pas être remise en cause.

Détaillé	La réponse peut exiger des déroulements pas à pas qui peut être demandés et donc acceptés.
Non monotone	Pour une meilleure compétence, il est possible de supprimer ou d'ajouter de la connaissance sans remise en compte du comportement global du système.
Sans limite	La connaissance se fait avec la grande capacité de stockage, il est donc toujours possible de stocker de la connaissance en cours de fonctionnement.
Fiabilité	le système expert doit être fiable et ne doit pas connaître des "failles" sinon il ne sera pas utilisé.
Compréhensible	le système doit être compréhensible, c'est-à-dire être capable d'expliquer les étapes du raisonnement lors de l'exécution. Le système expert doit avoir une capacité d'explication similaire à la capacité de raisonnement des experts humains.
Temps de réponse adéquat	le système doit être conçu de manière à pouvoir fonctionner dans un court laps de temps, comparable ou supérieur au temps mis par un expert humain pour atteindre un point de décision. Un système expert qui prend un an pour prendre une décision par rapport au temps d'un expert humain d'une heure ne serait pas utile.

Tableau1.1 : Quelques caractéristiques du système expert.

3.3.Architecture générale d'un système expert

Un système expert se compose d'une base de connaissances (base de fait et base de règle), d'un moteur d'inférences, mémoire de travaille et de différentes interfaces qui lui permettent de communiquer avec son environnement. Voir la [Figure1.2].

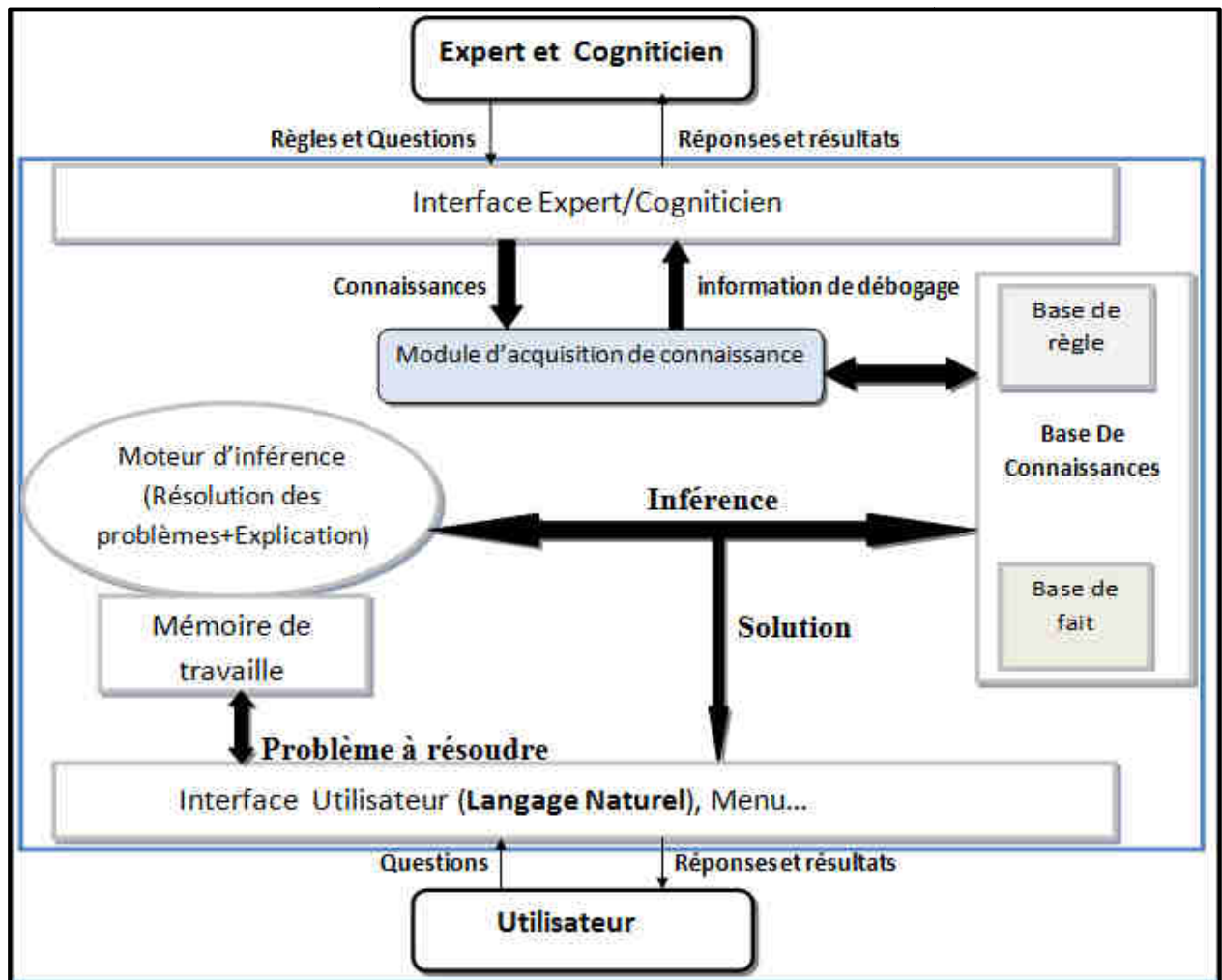


Figure1.2 : Architecture d'un Système expert.

A. La base de connaissances [6]

La base de connaissances est élaborée à partir de l'expertise d'un spécialiste. L'expertise elle-même s'obtient au cours d'un processus cognitif généralement long et encore mal connu de nos jours. C'est certainement à ce niveau que l'on peut situer la phase de création du savoir.

La base de connaissance est le noyau de système expert et elle diffère de la base de données classique. Dans les bases de données, on trouve des relations statiques entre les éléments de la base. La base de connaissance est créée par un cogniticien (ingénieur de connaissances). Ce dernier traduit les connaissances de l'expert humain en règles et stratégies. La base de connaissances se compose d'une base de faits et d'une base de règles.

- **Base de règles**

La base de règles 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**

La base de faits 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.

B. La Mémoire de travail

Contient les faits et les tâches spécifiques de résolution de problème en considération, appelée aussi module de base de données [2].c'est un espace de travail qui contient momentanément les règles d'inférences utilisées, les faits utilisés et les nouveaux faits inférés ainsi que toutes les données utilisées. Cet espace de travail joue le rôle d'une mémoire RAM dans un ordinateur.

C. Le moteur d'inférence

Le moteur d'inférence est un programme (Partie d'un système expert) qui utilise les règles définies dans la base de connaissances pour résoudre un problème particulier décrit par des faits [7]. Le MI est un mécanisme de contrôle qui organise les données de problème et cherche dans la base de connaissance les règles applicables pour ces données. A cause de l'augmentation de la popularité des systèmes experts, plusieurs moteurs d'inférences commerciaux sont apparus.

D. Les éléments complémentaires d'un SE

En plus de ces composants de base on peut trouver les modules supplémentaires pour faciliter la tâche de compréhension à l'utilisateur, on peut citer :

✓ Module d'interface utilisateur [8].

C'est un module de communication avec l'utilisateur d'une manière facile et flexible, sert à simplifier la communication, il peut utiliser la forme question-réponse, le menu, le langage naturel... etc.

✓ Module d'acquisition des connaissances

Une interface permettant à l'expert et au cogniticien d'alimenter, de mettre au point et de tester la base de connaissances (insertion, modification et suppression de parcelles de connaissances) Le cogniticien (ou l'ingénieur de la connaissance) est la personne chargée de soutirer les connaissances de l'expert et de modéliser ces connaissances dans un formalisme exploitable par le système.

✓ **Module d'explication [2].**

C'est un module qui sauvegarde les traces de l'exécution de système expert ainsi que toutes les informations du déroulement de processus de raisonnement afin de fournir ces informations aux utilisateurs de système expert. Alors le système expert doit être capable d'expliquer le raisonnement, par exemple, l'inférence de quelques conclusions durant le processus de déduction.

3.4. Les domaines d'application d'un SE

Il existe de nombreuses applications qui nécessitent l'utilisation d'un système expert pour résoudre des problèmes complexes. [Tableau1.2] décrit les domaines génériques des applications où le SE peut être appliqué. [W5].

Application	Description
Demain de conception	Conception d'objectif de caméra, conception automobile..etc
Systèmes de surveillance	Comparaison continue des données avec le système observé ou avec un comportement prescrit comme la surveillance des fuites dans les longs oléoducs.
Domaine de connaissances	Trouver des défauts dans les véhicules, les ordinateurs.
Diagnostic	Inférence des pannes en se basant sur les symptômes observés.
Finance/Commerce	Détection d'éventuelles fraudes, transactions suspectes, négociation boursière, programmation aérienne, programmation cargo.
Contrôle	Régulation d'un processus: faire une interprétation, diagnostic, surveillance, planification et pronostic.
Pronostic	Prédire les résultats d'une situation donnée.
Configuration	Assemblage propre des composants d'un système d'une manière correcte.
Interprétations	Explication de données observées.
Planification	élaborer ou modifier un plan d'action et Partager les actions pour l'obtention des résultats désirés.

Tableau1.2 : Quelques domaines d'application d'un système expert.

3.5. Quelques exemples et évolution de SE

Les systèmes experts ont fait leur apparition dans certains domaines et plusieurs sciences (Médecine, Mathématiques, Chimie, Biologie, etc.) Nous citons quelques exemples : [9].

DENDRAL, 1969 : chimie, recherche la formule développée d'un corps organique à partir de la formule brute et du spectrogramme de masse du corps considéré.

MYCIN, 1974 : médecine, système d'aide au diagnostic et au traitement de maladies bactériennes du sang.

AM, 1977 : mathématiques, proposition de conjectures, de concepts intéressants.

(500 règles)

MOLGEN, 1977: biologie, engendre un plan de manipulations génétiques en vue de construire une entité biologique donnée.

PROSPECTOR, 1978 : géologie, aide le géologue à évaluer l'intérêt d'un site en vue d'une prospection minière. (1600 règles)

CRYALIS, 1979 : chimie, recherche la structure de protéines à partir de résultats d'analyse cristallographique.

MUSCADET, 1984 : mathématiques, démonstration de théorèmes.

3.6. Les avantages de système expert

Les points forts d'un système expert sont : [10]

- ✓ Meilleur rendement que les experts humains.
- ✓ Lisibilités des connaissances, aspect plus déclaratif que procédural.
- ✓ Préservation de l'expertise.
- ✓ Facilité de mise à jour.
- ✓ Avantage des Systèmes d'Information en général.

3.7. Les inconvénients de système expert

Les deux principaux inconvénients des systèmes experts sont : [10] [W6].

- ✓ Ils créent le chômage parce qu'ils émulent les humains.
- ✓ Dans les systèmes experts, on fait inférence à des connaissances même si elles sont dépassées.
- ✓ Manque de flexibilité Efficacité limitée à un domaine spécifique.

4. Représentation des connaissances

4.1.Introduction

La puissance d'un système expert réside dans sa faculté à stocker et à manipuler la connaissance.

Le stockage de la connaissance nécessite sa modélisation et sa mise en œuvre informatique. La modélisation ou représentation des connaissances est l'un des problèmes clefs rencontrés dans les systèmes experts [11]. En effet, les modèles de la connaissance, doivent être suffisamment riches, pour pouvoir supporter les tenants du raisonnement humain.

❖ Qu'est-ce que la connaissance?

La compétence Ce qui permet la résolution de problèmes, En d'autres termes, La connaissance est définie par sa fonction, c.-à-d. se caractérise par ce qu'elle fait, non pas par son contenu structurel.

❖ Qu'est-ce que la représentation des connaissances?

La représentation des connaissances dans un domaine est signifie la modélisation des concepts humaines utilisé(les faits) pour résoudre des problèmes.

4.2.Type de représentation des connaissances

D'après la classification de J L Laurière [12]. Les connaissances sont classées par rapport à leurs natures de représentation : procédurales ou déclaratives. (Voir Figure3).

- **La représentation déclarative** : permet de spécifier un savoir indépendamment des contraintes et des méthodes d'utilisation. Elle permet de répondre à une question de "quoi ? ". La structure de contrôle est séparée des connaissances entrées sous formes de règles données en vrac. L'intérêt primordial est la modularité.
- **La représentation procédurale** : permet, de traiter des problèmes de type algorithmique c'est-à-dire complètement analysés et entièrement connus. Elle exprime un flot d'informations structurées et traduit "comment" transite la connaissance qui est une simulation du comportement réel.

Les représentations peuvent être regroupées dans les trois grandes classes suivantes :

Procédurales : incluant les automates finis et les programmes, **déclaratives** : comprenant le calcul des prédicats et les règles de production, **structurées** : réseaux sémantiques, frames, schémas, scripts, objets.

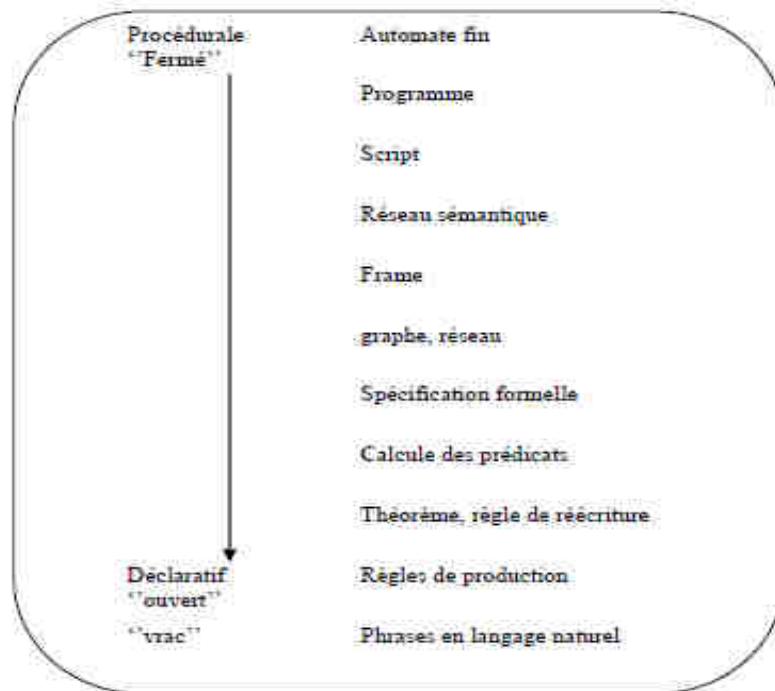


Figure1.3 : Classification de J.L. Laurière [12].

A. Représentation procédurale

Cette représentation mélange des données limitées aux faits et contrôle. Ce mode de représentation de connaissance est peut utilisable car il est difficile à modifier et à entendre de l'évolution de la connaissance.

- Les automates finis

Un automate est une représentation procédurale de la connaissance utilisée pour représenter des protocoles ou représenter des plannings d'actions. Une action se trouve dans un nœud et un arc représente la règle à entreprendre (à exécuter) ou prédicat pour passer d'un état à un autre.

- Les programmes

Un programme est une suite finis d'instructions ordonnées, exécutables par ordinateur.

Les programmes sont souvent représentés par des procédures ou des morceaux de programmes à exécuter en cas de besoin par une série d'appels. Il existe deux sortes d'appels : appel directe, attachement procédural.

B. Représentation logique [13].

- **Logique d'ordre 0**

On parle indifféremment de logique booléenne, de logique des propositions: c'est la plus simple des logiques, appelée la logique d'ordre 0.

Dans cette logique, les formules sont appelées des propositions.

- On définit le calcul propositionnel, comme on définit le calcul arithmétique.

- Les formules sont construites à l'aide de variables issues d'un ensemble dénombrable et des symboles { }.

Exemple : Il n'y a pas d'animal qui vole = Aucun animal ne vole.

$\neg (\exists X \text{ animal}(X) \Rightarrow \text{vole}(X)).$

$\forall X \text{ animal}(X) \Rightarrow \neg \text{vole}(X).$

▪ Logique d'ordre 0+

D'un point de vue sémantique, cela revient à dire que l'on considère les fonctions et prédicats comme des objets à part entière, au même titre que, par exemple, un nombre entier. On s'autorisera ainsi, d'une part, à quantifier les prédicats et fonctions et, d'autre part, à donner des fonctions ou des prédicats en arguments d'autres fonctions et prédicats. Néanmoins, on pourra se doter d'un système de typage qui restreindra le genre d'objet qui pourra être donné en tant que tel ou tel argument de tel ou tel prédicat ou telle ou telle fonction.

Un prédicat d'ordre supérieur est un prédicat qui prend comme argument un ou plusieurs autres prédicats. De manière générale, un prédicat d'ordre n prend comme argument un ou plusieurs prédicats d'ordre n-1, avec $n > 1$. La même chose est valable pour les fonctions d'ordre supérieur.

Exemple : $\underbrace{\text{SI } \text{âge} \geq 18}_{\text{Prémisse}} \text{ ALORS } \underbrace{\text{statut} = \text{« majeure »}}_{\text{Conclusion}}$

▪ Logique d'ordre 1(logique des prédicats)

C'est la logique du calcul des prédicats. Définition du prédicat : Expression logique dont la valeur peut être vraie ou fausse selon la valeur des arguments, c'est-à-dire une fonction qui renvoie "vrai" ou "faux".

La logique d'ordre 1 contient :

- ✓ des variables substituables = variables d'individu,
- ✓ des quantificateurs : Pour tout, Quelque soit, ...etc.

Exemples :

-Tous les lions sont féroces

Lion (X) \Rightarrow X est un lion

Féroce(X) \Rightarrow X est féroce

$(\forall X) (\text{Lion}(X) \Rightarrow \text{Féroce}(X)).$

C. Règles de production

Ils utilisent de l'expertise et de la connaissance pour en arriver à résoudre des problèmes qui nécessitent de l'intelligence humaine. Ces deux éléments sont représentés sous la forme de règles dans l'ordinateur [14]. Une règle est un énoncé conditionnel qui lie ensemble des conditions à des actions ou résultats. Elles sont sous la forme : **SI** Prémisse(s) **ALORS** Conséquence(s).et permettent de formuler les énoncés conditionnels que nous retrouvons dans la base de connaissances.

Le langage de description des règles peut être plus ou moins complexe, comporter des conjonctions, des disjonctions, des négations ... [15].

Exemple :

Les règles :

- R1 : **si** aucune réaction aux actions du clavier **et** les lettres tapées ne s'affichent pas sur l'écran **Alors** problème dans le clavier
- R2 : **si** Ecran bleu **et** ralentissement de l'ordinateur **Alors** problème dans la ram

D. Les réseaux sémantiques

❖ Définition

Les réseaux sémantiques constituent une classe de structures de données représentables facilement dans un ordinateur. Ils sont utilisés pour cette raison en intelligence artificielle comme des modes de représentation des connaissances. [16]

Donc, Les réseaux sémantiques sont des graphes dont les nœuds sont des entités reliées entre elles par des arcs orientés et étiquetés, signifiant des relations d'appartenance, causales, spatiales, fonctionnelles, etc. L'organisation des données est bien adaptée à la description et permet la sélection des faits pertinents dans les mécanismes de déductions.

Exemple :

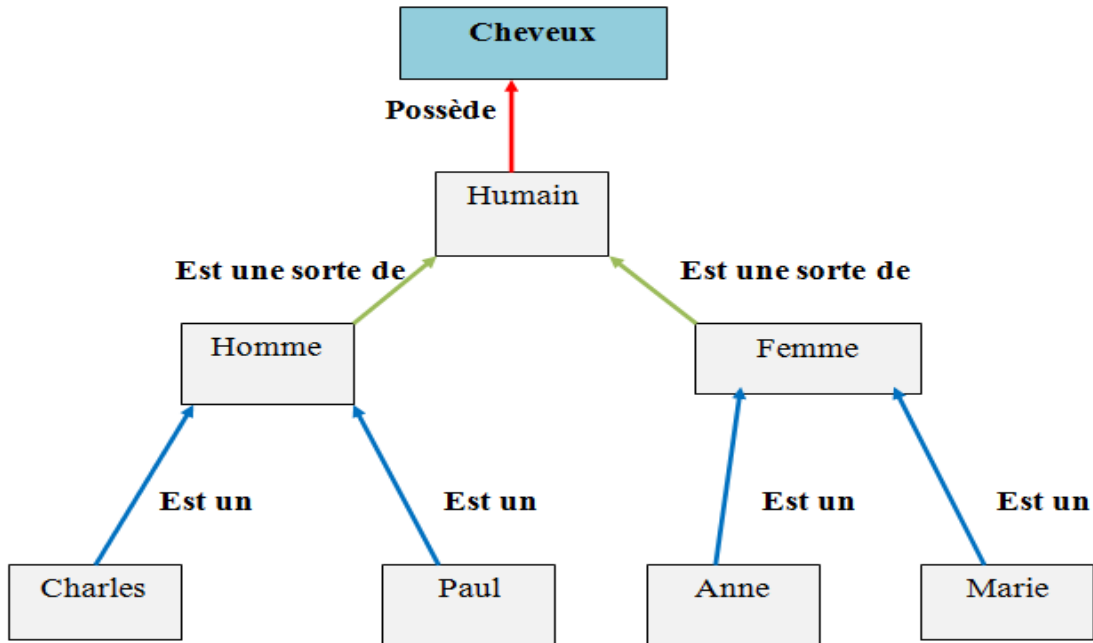


Figure1.4 : Représentation d'un réseau sémantique.

E. Les réseaux de neurones

❖ Définition.

Un réseau de neurones est une fonction paramétrée qui est la composition d'opérateurs mathématiques simples appelés neurones formels (ou plus simplement neurones) pour les distinguer des neurones biologiques. [13].

❖ Les neurones

Un neurone est une fonction algébrique non linéaire, paramétrée, à valeur bornée de variables réelles appelées *entrées*. Un neurone réalise une fonction non bornée

$$y=f\{x_1, x_2, \dots, x_n; w_1, w_2, \dots, w_n\} \text{ où les :}$$

{X} _i : sont les entrées

{W} _j : sont des poids

F: est la fonction d'activation

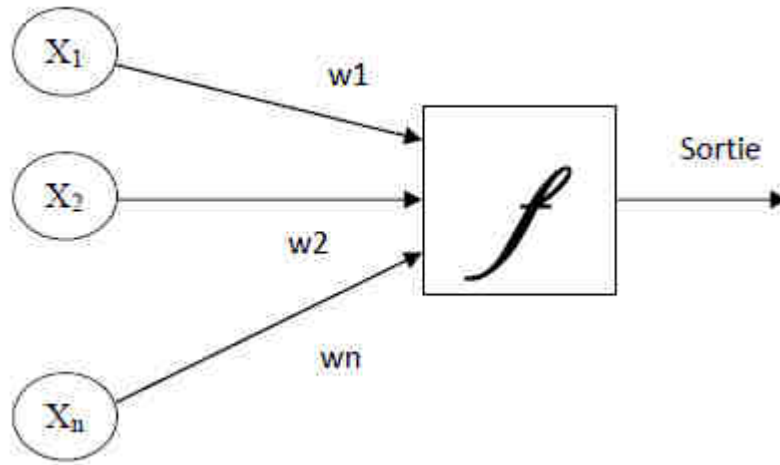


Figure1.5 : Représentation d'un réseau de neurones [13].

F. Les objets structurés.

Il est impossible de parler de la Programmation Orientée Objet sans parler d'objet, bien entendu. Tâchons donc de donner une définition aussi complète que possible d'un objet. Un objet est avant tout une structure de données. Autrement, il s'agit d'une entité chargée de gérer des données, de les classer, et de les stocker sous une certaine forme. En cela, rien ne distingue un objet d'une quelconque autre structure de données. La principale différence vient du fait que l'objet regroupe les données et les moyens de traitement de ces données. Un objet rassemble de fait deux éléments de la programmation procédurale. [W7].

- ✓ **Les champs** Les champs sont à l'objet ce que les variables sont à un programme : ce sont eux qui ont en charge les données à gérer. Tout comme n'importe quelle autre variable, un champ peut posséder un type quelconque défini au préalable : nombre, caractère... ou même un type objet [W7]
- ✓ **Les méthodes** Les méthodes sont les éléments d'un objet qui servent d'interface entre les données et le programme. Sous ce nom obscur se cachent simplement des procédures ou fonctions destinées à traiter les données.

Exemple :

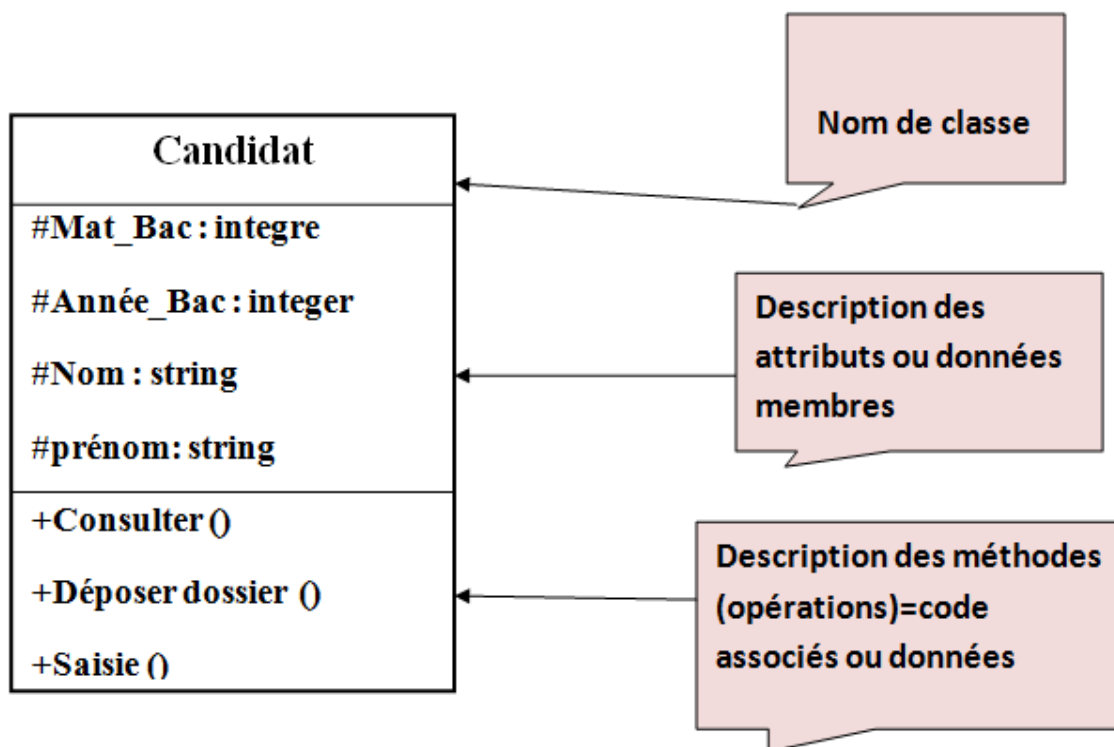


Figure1.6 : Représentation de la classe candidat.

Les trois fondamentaux dans la structure d'objet sont :

- **Encapsulation**

Est déjà une réalité dans les langages procéduraux (comme le Pascal non objet par exemple) au travers des unités et autres bibliothèques, il prend une toute nouvelle dimension avec l'objet. En effet, sous ce nouveau concept se cache également un autre élément à prendre en compte : pouvoir masquer aux yeux d'un programmeur extérieur tous les rouages d'un objet et donc l'ensemble des procédures et fonctions destinées à la gestion interne de l'objet, auxquelles le programmeur final n'aura pas à avoir accès.

L'encapsulation permet donc de masquer un certain nombre de champs et méthodes tout en laissant visibles d'autres champs et méthodes [W7].

- **L'héritage [W8]**

Est un principe propre à la programmation orientée objet, permettant de créer une nouvelle classe à partir d'une classe existante. Le nom d'héritage (pouvant parfois être appelé dérivation de classe) provient du fait que la classe dérivée (la classe nouvellement créée) contient les attributs et les méthodes de sa superclasse (la classe dont elle dérive). L'intérêt majeur de l'héritage est de

pouvoir définir de nouveaux attributs et de nouvelles méthodes pour la classe dérivée, qui viennent s'ajouter à ceux et celles héritées.

Exemple

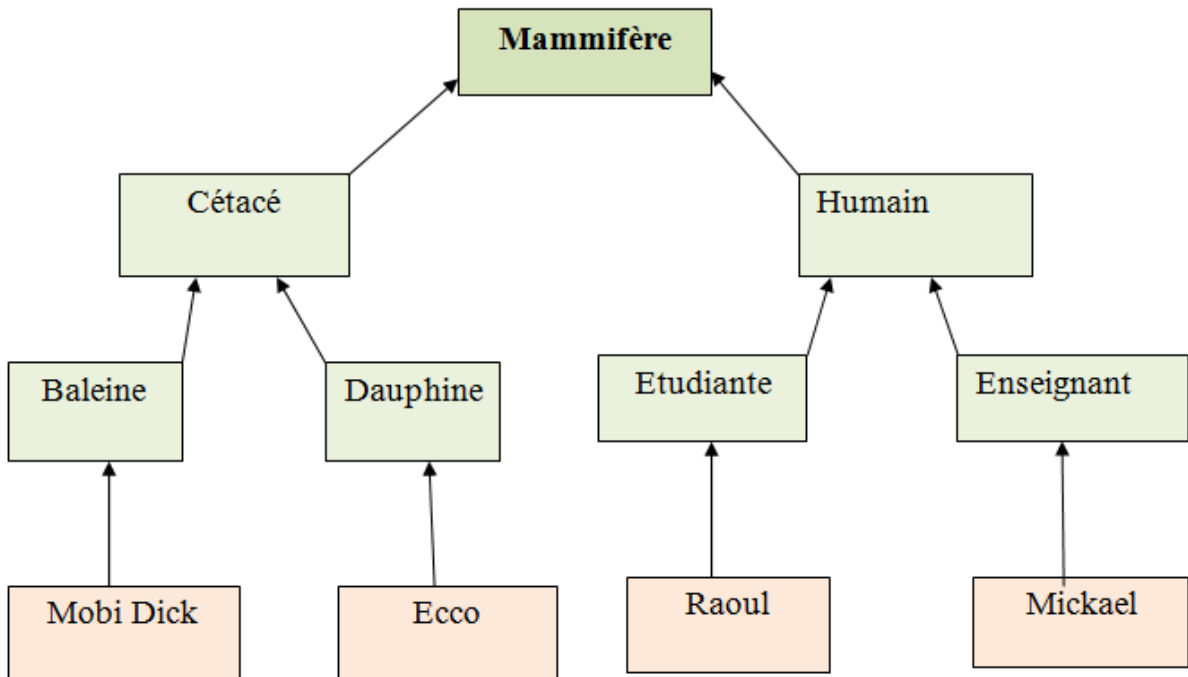


Figure1.7 : Exemple d'héritage.

▪ Polymorphisme

Le polymorphisme en informatique est l'idée d'autoriser le même code à être utilisé avec différents types, ce qui permet des implémentations plus abstraites et générales [W9].

Il signifie **“qui peut prendre plusieurs formes”**. Cette caractéristique est un des concepts essentiels de la programmation orientée objet. Alors que l'héritage concerne les classes (et leur hiérarchie), le polymorphisme est relatif aux méthodes des objets.

Exemple :

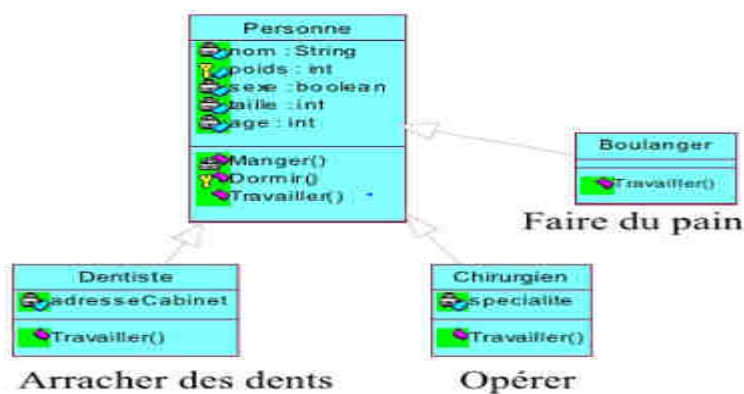


Figure1.8 : Exemple de polymorphisme [W10].

5. Structure Caractéristique d'un moteur d'inférence

5.1. Le cycle de travail de moteur d'inférence

Le moteur d'inférence (appelé parfois moteur de résolution) est le cœur d'un système expert. Celui-ci peut être indépendant du domaine d'application et il doit être capable d'exploiter la base des connaissances afin de résoudre les problèmes posés par les utilisateurs. En effet, il doit être capable de voir quelles sont les règles applicables au vu de l'état courant de la base des faits et des faits à établir (exprimant les problèmes /questions posés par l'utilisateur). Ce processus de détection des règles applicables fait partie du cycle de travail d'un moteur d'inférence et particulièrement de sa phase d'évaluation [17]

A. Phase d'évaluation [W11]

✓ Cette phase se déroule en trois étapes :

- **La sélection ou restriction**

Qui a pour objet de trier et de rassembler en un sous-ensemble, les faits et les règles de la base de connaissances qui méritent plus d'attention que d'autres.

- **Le filtrage**

Le moteur d'inférence compare la partie prémisse des règles sélectionnées avec les faits de la base de faits pour déterminer l'ensemble des règles applicables.

- **La résolution de conflits**

Se concrétise par le choix de la règle à appliquer. Cette étape, de loin la plus importante, manifeste également une stratégie qui peut être très simple et sans rapport avec le contexte (la première règle de la liste, la moins complexe ; la moins utilisée...) ou plus complexe en tenant compte du contexte (la plus prometteuse, la plus fiable,...).

B. Phase d'exécution [W11]

Cette phase consiste à appliquer la règle choisie et mettre à jour la base de faits BDF.

✓ La [Figure1.9] illustre le cycle de base de moteur d'inférence à travers les deux phases :

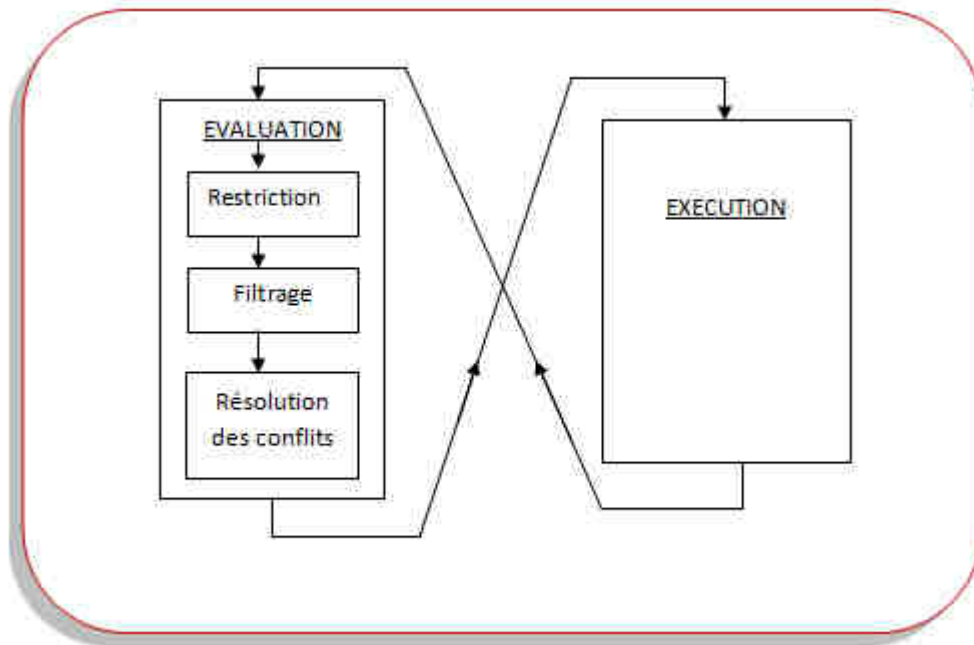


Figure1.9 : Représentation de cycle de travail de moteur d'inférence.

5.2.Mode de résonnement

Généralement deux modes de raisonnement sont utilisés, Le chaînage avant et le chaînage arrière, et un troisième qui combine les deux modes appelé le chaînage mixte.

A. Chainage avant (raisonnement guidé par les données)

Un moteur d'inférence fonctionne dans ce mode lorsque les faits de la base de faits représentent des informations dont la valeur de vérité a été prouvée. C'est-à-dire que ce mode de fonctionnement va des faits vers les buts. [17]

- **Algorithme du chaînage avant :**

BF (base de faits), BR (base de règles), F (proposition à vérifier)

DEBUT

TANT QUE F n'est pas dans BF **ET QU'**il existe dans BR une règle applicable **FAIRE**

Prendre la première règle applicable R

BR = BR - R (désactivation de R)

BF = BF union conclusion(R) (déclenchement de la règle R, sa

Conclusion est rajoutée à la base de faits)

FIN TANT QUE

SI F appartient à BF **ALORS**

F est établi (succès)

SINON

F n'est pas établi (échec)

FIN

Exemple :

BF_(initial) = {a, b} et

BR = {

R1 : *si a alors c*

R2 : *si a et d alors e*

R1 : *si e alors f*

R2 : *si b et c alors d*} alors on peut déduire c, d, e, f → BF_(final) = {a, b, c, d, e, f}

B. Chainage arrière (raisonnement guidé par le but)

A l'inverse de chainage avant, le chainage arrière est utilisé pour propager des conclusions vers les conditions de la règle. Le chainage arrière est très utile dans le cas où les résultats (but) sont connus et ne sont pas très nombreux. Dans ce cas, le but est spécifié et le moteur d'inférence cherche à trouver les conditions nécessaires pour arriver à ce but [18].

- **Algorithme du chainage arrière :**

```

Fonction chainageArrière

Paramètres : in BR, in BF, in listeButs.

SI est Vide(listeButs) ALORS

RES ← SUCCES

SI

SI demBut (premier(listeButs)) ALORS

RES ← chainageArrière(suite(listeButs))

SI

RES ← ECHEC

FIN SI

FIN SI

Retourner RES

```

Exemple

BF_(initial) = {a, b} et

Le but = f

BR = {

alors on peut déduire :



R1 : si a alors c

R2 : si a et d alors e

R1 : si e alors f

R2 : si b et c alors e}

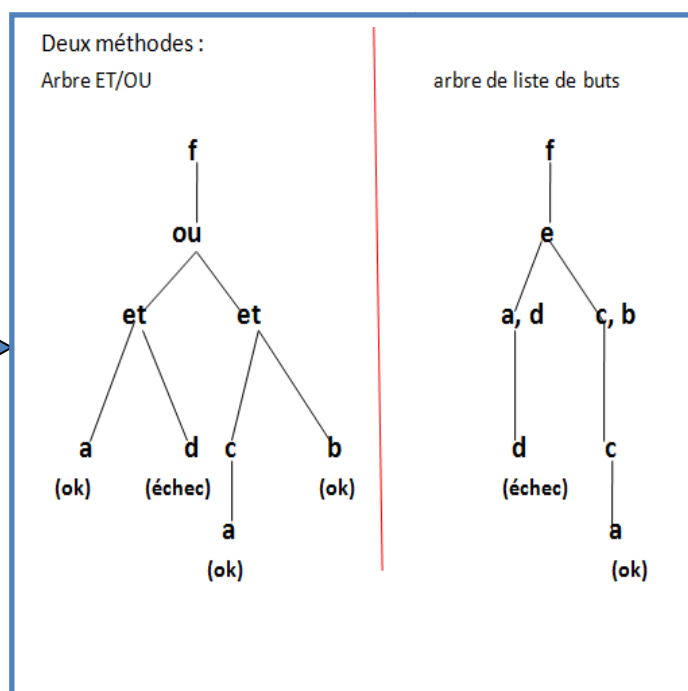


Figure1.10: Représentation du résultat de l'exemple.

La [Figure1.10] représente le résultat de notre exemple.

C. Chainage mixte [19]

Le chainage mixte combine entre l'utilisation des chinages avant et arrière, les deux chinages utilisés ensemble pour la résolution automatique des problèmes.

6. Stratégie de recherche

Au cours des cycles de recherche d'un MI, on développe un arbre de recherche dans lequel chaque niveau correspond à l'ensemble des règles applicables (ensemble de conflits) Chaque règle déclenchée crée une nouvelle situation et de nouvelles règles à invoquer.

6.1.La Recherche en profondeur d'abord

Dans ce cas on s'enfonce dans l'arbre de décision en passant d'état en état jusqu'à ce que le chemin se termine et cherche à explorer au maximum une possibilité en appliquant les règles de façon à obtenir le plus de détails

- Chaînage arrière : ajout du retour-arrière
- Chaînage avant : imposition d'une limite de recherche (éviter voie sans issue) [20].

6.2.La Recherche en largeur d'abord

Dans ce cas on parcourt à l'horizontal un niveau de l'arbre de décision avant d'aller au suivant et on commence par explorer toutes les possibilités présentes avant d'entrer dans les détails. [20] .

6.3.La Recherche en profondeur limité

La recherche en profondeur limitée combine les deux approches précédentes : profondeur d'abord et largeur d'abord.

- L'arbre des états est découpé en k niveaux.
- Une recherche en profondeur est effectuée dans chaque tranche avant de passer à la tranche inférieure. [20].

6.4.La Recherche heuristique

Des connaissances spécifiques sur le problème à traiter sont exploitées pour trouver une solution.

La connaissance du problème à traiter prend la forme d'une fonction d'évaluation qui mesure la « promesse » de se trouver sur un chemin intéressant. Une heuristique garantit une bonne solution mais pas nécessairement la solution optimale. [20].

7. Stratégie de contrôle

7.1. Régime de contrôle

- **Irrévocable** : L'application d'une règle dans un cycle du MI n'est jamais remise en cause et on n'opère pas de backtracking. S'il n'y a plus de règles à appliquer. Le MI s'arrête et signale un échec sans faire retour en arrière.
- **Tentative**: Ce régime peut remettre en cause des règles déjà appliquées si elles n'ont pas abouti, et faire un backtracking en retirant aussi les faits qui en étaient déduits.

7.2. Critère de monotonie

Un moteur d'inférence MI peut être :

- **Monotone**:

En régime monotone: le MI ne fait qu'ajouter des faits à la BDF et n'élimine jamais une règle de BDR.

- **Non monotone** :

En régime non monotone: le MI peut en cas de retour arrière par exemple retrancher de la BDF un fait précédemment ajouté

8. Conclusion

Nous concluons que le domaine de l'intelligence artificielle c'est un domaine très large, car il englobe tous les domaines parmi eux, les systèmes expert. Ces derniers sont des outils très important qui offrent un bon environnement pour implémenté l'expertise humaine et implanté dans tous les domaines et surtout le domaine informatique (notre sujet).

Comme nous avons besoin d'un outil pour nous aider dans la maintenance informatique, Nous avons choisi de concevoir un système expert intelligent qui nous aide à détecter les différentes pannes, dans le chapitre suivant nous présenterons la conception de notre système.

Chapitre 2 : conception de l'application

1. Introduction

Le développement de tout système expert passe par un cycle de vie allant de l'expression des besoins à l'exploitation et l'utilisation d'un produit final. Dans ce chapitre, nous nous concentrerons sur la phase de conception, qui est l'étape la plus importante pour obtenir un système expert valide et sûr, capable de diagnostiquer des défauts et des pannes d'un micro ordinateur soit matériel ou logiciel, que nous avons nommé SEDIP (Système Expert Diagnostic Pannes).

Il existe plusieurs méthodes de conception et de modéliser notre système (SEDIP), parmi lesquelles nous avons choisi la modélisation UML (Unified Modeling Language) qui se présente comme un métalangage par rapport aux langages de programmation, il repose sur une notation graphique indépendante des langages de programmation qui donne à chaque concept objet une représentation particulière, au moyen de neuf types de diagrammes qui sont autant de vues mutuellement cohérentes sur le même modèle.

Nous apprendrons à connaître plus sur cette dernière à travers l'étude que nous entreprendrons dans cette partie, où nous étudions d'abord l'architecture de notre système, et Ensuite la conception UML.

2. Architecture du système

2.1. Description générale du système SEDIP :

L'architecture de notre SEDIP passe par quatre étapes de base: premièrement, l'étape d'acquisition et gestions des connaissances, deuxièmement, l'étape d'effectuer la recherche, troisièmement, l'étape de traitement et raisonnement, et quatrièmement, l'étape de communication de solution. En plus de cela ce dernier est composé de trois éléments principaux: la base de connaissance (base de faits et base de règle), le moteur d'inférence, les interfaces graphique, et deux acteurs l'expert et l'utilisateur, Tout est montré dans la [Figure2.1].

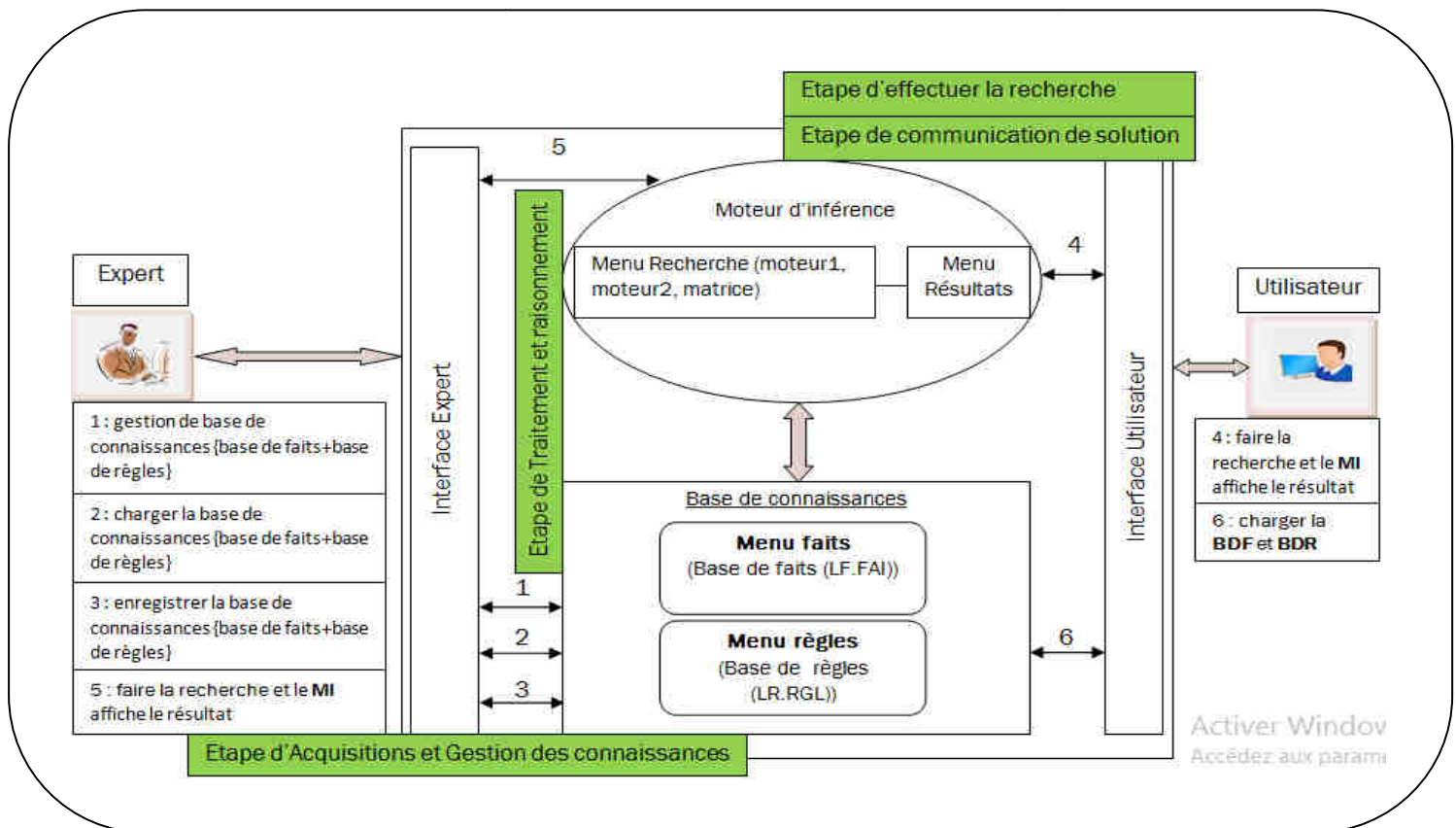


Figure2.1: L'architecture de notre système.

2.2. Description détaillée du système SEDIP

Dans cette partie on va détailler les étapes de notre système SEDIP :

A. L'étape d'acquisition des connaissances :

Dans cette étape l'expert crée d'abord un projet qui contient une base de faits et une base de règles et un moteur d'inférence(MI) pour le raisonnement et la recherche, et ensuite il alimente la base de connaissances(BDF , BDR) et donne les détails nécessaires sur les types des pannes(les faits) soit matériel ou logiciel.

On passe maintenant à la gestion de cet dernier, Où l'expert développe et teste la base de connaissances (BDF, BDR). C'est à dire faire les opérations suivantes sur la BDF : l'ajout, la suppression, la modification. Et l'ajout, la suppression sur la BDR.

En plus de ces opérations, il peut également sauvegarder, charger un projet avec l'extension .PRJ c.à.d. sauvegarder, charger la base faits avec l'extension .FAI et la base de règle avec l'extension .RGL, Voir la [Figure2.2] pour plus de détails. Et tout cela se fait via l'interface graphique de l'expert.

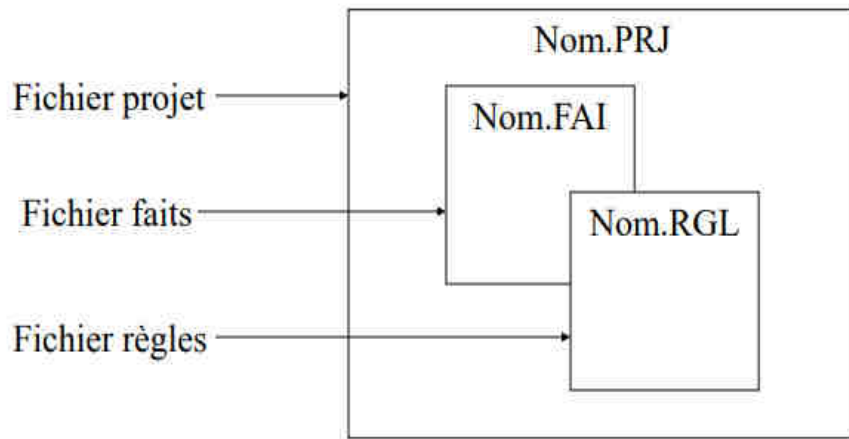


Figure2.2 : Représentation de stockage des données.

B. L'étape d'effectuer la recherche :

L'utilisateur peut effectuer des recherches par l'intermédiaire du moteur d'inférence (menu de recherche), qui répondra à ses questions en fonction des règles créées (menu de résultats), c.à.d. il sélectionne d'abord le type de l'opération : le chaînage avant (moteur1) ou le chaînage arrière (moteur2) et après introduit les données à traiter.

C. L'étape de traitement et raisonnement :

Le MI contient de deux menus : le premier est le menu de recherche c'est l'espace dans lequel l'utilisateur introduit ses données sur son problème, le deuxième est le menu de résultat c'est l'espace dans lequel le résultat est montré à l'utilisateur.

Après que l'utilisateur avoir entré son choix de l'opération (chaînage avant ou arrière) et les symptômes, le MI enregistre ces données (les faits initiaux) à traiter dans la première ligne de la matrice, puis commence le traitement, D'abord, il lire et parcourt la base de connaissances (BDF, BGR), et Ensuite, parmi les règles déclenchable, il Choisit la règle qui va être appliquée en premier selon les stratégies de recherche qui choisit par l'utilisateur nous avons mentionné précédemment.

D. L'étape de communication de solution:

Le moteur d'inférence (menu de résultats) communique à l'utilisateur la meilleure solution au problème posé par l'intermédiaire de l'interface utilisateur et attend des nouvelles instructions.

3. Conception UML

3.1. Introduction

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. UML a une dimension symbolique et ouvre une nouvelle voie d'échange de visions systémiques précises. Ce langage est certes issu du développement logiciel mais pourrait être appliqué à toute science fondée sur la description d'un système. Dans l'immédiat, UML intéresse fortement les spécialistes de l'ingénierie système. UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la spécification jusqu'au codage. Dans ce cadre, un concept appartenant aux besoins des utilisateurs projette sa réalité dans le modèle de conception et dans le codage. Le fil tendu entre les différentes étapes de construction permet alors de remonter du code aux besoins et d'en comprendre les tenants et les aboutissants. En d'autres termes, on peut retrouver la nécessité d'un bloc de codes en se référant à son origine dans le modèle des besoins [21].

3.2. Les diagrammes de notre système

A. Diagramme de cas d'utilisation :

❖ Définition

Les cas d'utilisation ont été définis initialement par Ivar Jacobson en 1992 dans sa méthode OOSE. Les cas d'utilisation constituent un moyen de recueillir et de décrire les besoins des acteurs du système. Ils peuvent être aussi utilisés ensuite comme moyen d'organisation du développement du logiciel, notamment pour la structuration et le déroulement des tests du logiciel.

Un cas d'utilisation permet de décrire l'interaction entre les acteurs (utilisateurs du cas) et le système. La description de l'interaction est réalisée suivant le point de vue de l'utilisateur. La représentation d'un cas d'utilisation met en jeu trois concepts : l'acteur, le cas d'utilisation et l'interaction entre l'acteur et le cas d'utilisation. [22].

❖ Identification des acteurs :

Notre système sera utilisé par trois acteurs sont :

-L'expert (Technicien supérieur en maintenance informatique) :

Il est la personne responsable de la gestion du système c.à.d. gérer la base de faits (ajouter, supprimer, renommer,) et base de règles (ajouter, supprimer), et ensuite il peut charger, enregistrer un projet (BDF, BDR).

Remarque : Puisque l'expert est l'administrateur du système, il peut faire tous les rôles des utilisateurs (par ex : il faire des recherche).

-L'utilisateur :

C'est une personne qui cherche un diagnostic sur la panne, en introduisant les faits (données) au système, Cette personne pourrait être soit un technicien, technicien supérieur en maintenance informatique ou bien un gérant de parc informatique ou enfin une personne désirant apprendre sur les pannes d'ordinateur.

- SEDIP (système expert) :

- ✓ Le système fournit à l'expert et l'utilisateur une interface qui lui permet d'accéder à l'application.
- ✓ Le système devra reconnaître l'expert grâce à son login et mot de passe (authentification) afin de lui permettre de manipuler la base de règles et la base de faits.
- ✓ le système déduit les pannes relatives aux faits insérés, C.-à-d. infère les résultats.

Chaque acteur joue son rôle, Nous montrons au dessous les cas d'utilisation de chacune, Voir la [Figure2.3].

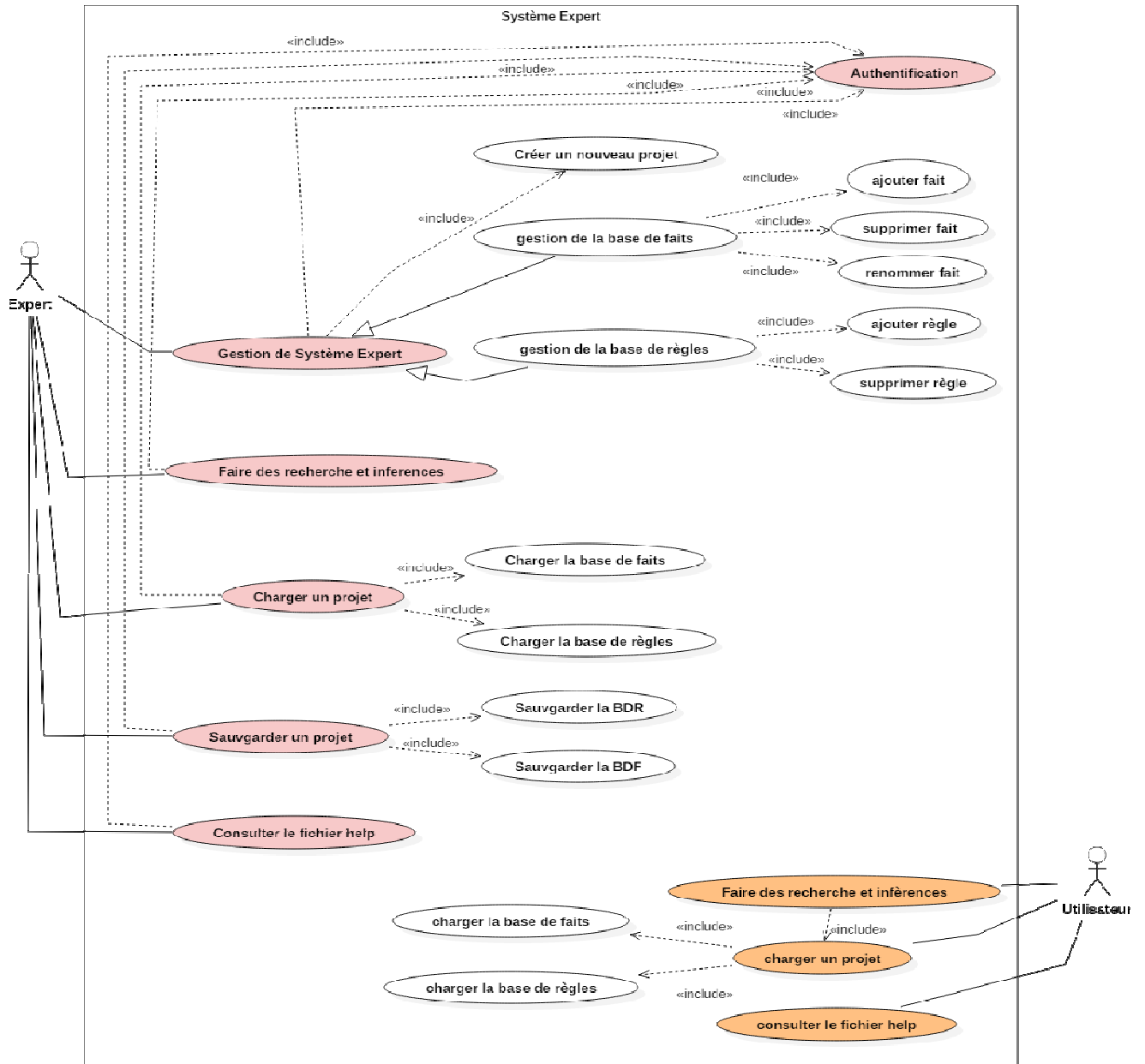


Figure2.3 : Diagramme de cas d'utilisation Expert _ Utilisateur.

❖ Description détaillée des cas d'utilisation

À chaque cas d'utilisation doit être associée une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs. [22].

- **Coté « Expert »**

<u>Acteur principale : Expert.</u>	
<u>Les cas d'utilisation :</u>	<u>Type de relation :</u>
<p>-Gestion de système expert : cette fonctionnalité permet au l'expert de gérer la base de connaissance (BDF, BDR), De plus, tout d'abord, l'expert doit créer un nouveau projet.</p> <ul style="list-style-type: none"> • gestion de base de faits : dans ce cas, l'expert peut : <ul style="list-style-type: none"> ✓ Ajouter un fait. ✓ Supprimer un fait. ✓ Renommer un fait. • gestion de base de règles : dans ce cas, l'expert peut : <ul style="list-style-type: none"> ✓ Ajouter une règle. ✓ Supprimer une règle. ✓ Ajouter ou supprimer une liste de faits d'une règle. ✓ Ajouter ou supprimer une liste de conséquences de faits d'une règle. ✓ Supprimer tous les règles. 	<p>-la relation de communication (association) entre l'expert et le cas d'utilisation gestion de système expert.</p> <p>-la relation de généralisation : nous avons le cas général le cas d'utilisation gestion de système expert.</p> <p>-la relation de spécification : nous avons les deux cas particulier le cas d'utilisation gestion de la base de faits et le cas d'utilisation gestion de la base de règles</p> <p>-la relation d'inclusion (c'est une sous fonction) : la relation d'inclusion est impérative et donc symétrique. Il est nécessaire d'appliquer toutes les opérations (mentionnées précédemment) aux deux bases.</p>
<p>-Faire des recherches et inférences : dans ce cas, l'expert peut :</p> <ul style="list-style-type: none"> ✓ Choisir le type de recherche (chainage avant /chainage arrière). ✓ Entrer des faits (symptômes) initiaux. (les deux actions sont faites dans le menu de recherche). 🚦 le système(SEDIP) infère les résultats. 🚦 Le système(SEDIP) affiche les résultats dans le Menu de résultats. 	<p>-la relation de communication (association) entre l'expert et le cas d'utilisation faire des recherches et inférences.</p>
<p>-Charger/Enregistrer un projet : dans ce cas, l'expert peut :</p> <ul style="list-style-type: none"> ✓ Charger/Enregistrer la base de faits. ✓ Charger/Enregistrer la base de règles. 	<p>-la relation de communication (association) entre l'expert et les deux cas d'utilisation charger/Enregistrer un projet.</p> <p>-la relation d'inclusion (c'est une sous fonction) : la relation d'inclusion est impérative et donc symétrique.</p>

	C'est obligatoirement de charger et enregistrer la base de faits et la base de règles.
- Consulter le fichier help.	-la relation de communication (association) entre l'expert et le cas d'utilisation consulter le fichier help.
Pré-condition : Au préalable l'expert doit s'authentifier.	

Tableau2.1 : Description détaillé pour les cas d'utilisation coté Expert.

▪ **Coté « Utilisateur »**

Acteur principale : Utilisateur.	
Les cas d'utilisation :	Type de relation :
<p>-Faire des recherches et inférences : dans ce cas, l'utilisateur peut :</p> <ul style="list-style-type: none"> ✓ Choisir le type de recherche (chainage avant /chainage arrière). ✓ Entrer des faits (symptômes) initiaux. (les deux actions sont faites dans le menu de recherche). 🚦 le système(SEDIP) infère les résultats. 🚦 Le système(SEDIP) affiche les résultats dans le Menu de résultats. 	<p>-la relation de communication (association) entre l'utilisateur et le cas d'utilisation faire des recherches et inférences.</p>
<p>-Charger un projet : dans ce cas, l'utilisateur peut :</p> <ul style="list-style-type: none"> ✓ Charger la base de faits. ✓ Charger la base de règles. 	<p>-la relation de communication (association) entre l'utilisateur et les deux cas d'utilisation charger un projet.</p> <p>-la relation d'inclusion (c'est une sous fonction) : la relation d'inclusion est impérative et donc symétrique.</p> <p>pour charger un projet, il obligatoirement de charger son base de fait, et son base de règle.</p>

- Consulter le fichier help.	-la relation de communication (association) entre l'utilisateur et le cas d'utilisation consulter le fichier help .
Pré-condition : Avant d'effectuer la recherche, l'utilisateur doit d'abord charger le projet.	

Tableau2.2 : Description détaillé pour les cas d'utilisation coté Utilisateur.

✚ Cas dans lesquels le système ne fonctionne pas :

- Fichier invalide (soit .FAI ou .RGL ou .PRJ).
- La base de faits ou règles est vide.
- Faits déjà existe (ex : Gestion de faits : ajouter un fait qui existe déjà).
- Règles déjà existe (ex : Gestion de règles : ajouter une règle qui existe déjà).
- Faits n'existe pas (ex : Gestion de faits : supprimer/renommer un faits qui n'existe pas)
- Règles n'existe pas (ex : Gestion de règles : supprimer une règle qui n'existe pas)

B. Diagramme de séquence :

❖ Définition

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML, comme les diagrammes de collaboration mais ils ne rendent pas compte du contexte des objets de manière explicite. [23].

➤ Pour chaque cas d'utilisation on à spécifier un diagramme de séquence :

▪ Coté « Expert »

1.1.Diagramme de séquence de cas d'utilisation(Authentification):

Si l'expert veut entrer dans sa session, il se dirige d'abord vers sa page d'authentification, où il saisit son nom d'utilisateur et son mot de passe. Une fois que les deux champs sont remplis le système effectuera une vérification sur les informations insérées si les données sont correctes alors le système lui permettra d'accéder directement à son interface, dans le cas contraire, il pourra refaire l'insertion à chaque fois que login ou le mot de passe seront erronés.

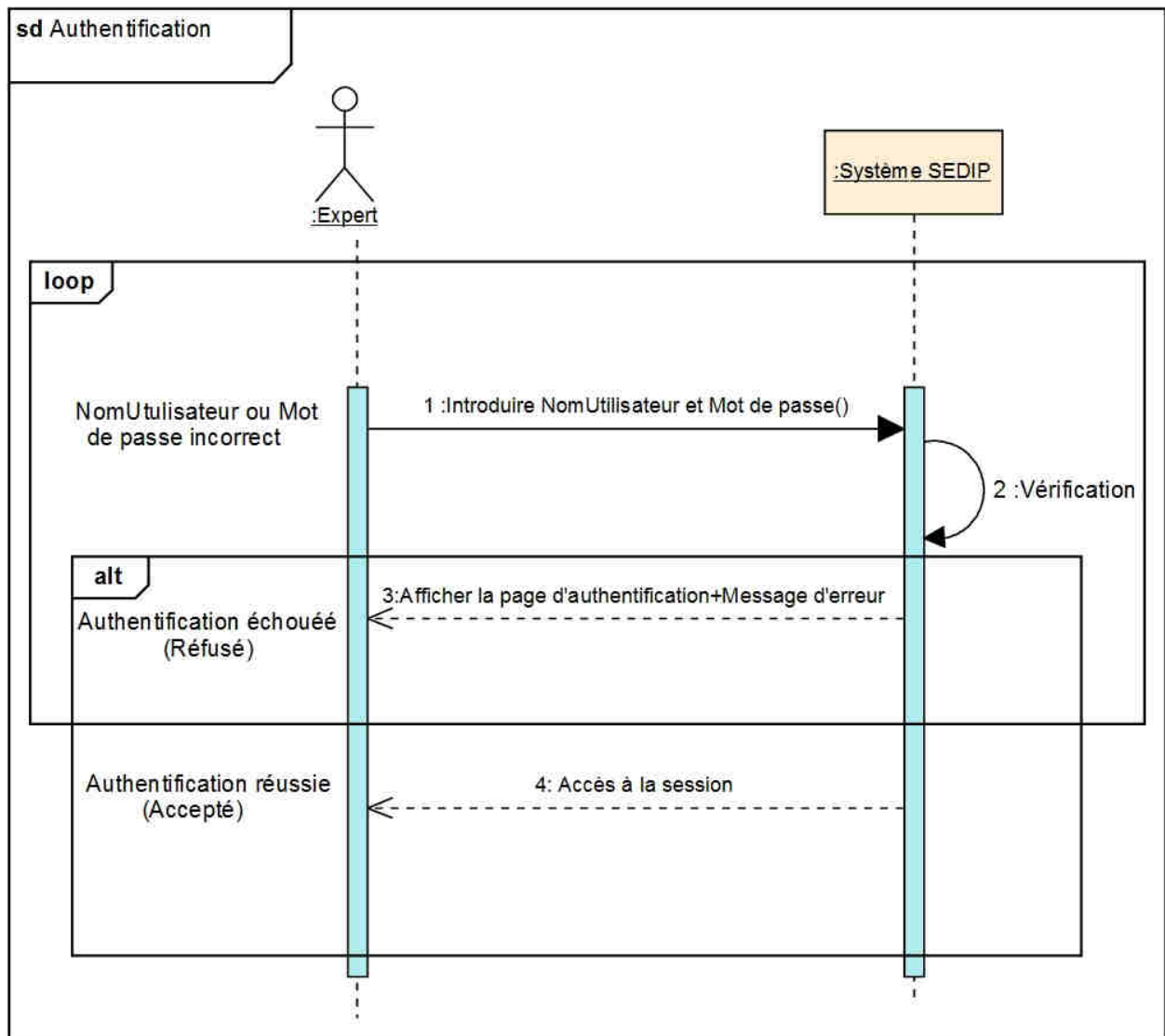


Figure2.4 : Diagramme de séquence de cas d'utilisation (Authentification).

1.2. Diagramme de séquence de cas d'utilisation (gestion de la base de faits):

Dans ce cas, l'expert entre dans sa session afin de gérer la base de faits, où il sélectionne d'abord l'opération (soit l'ajout ou la suppression ou renommer), et pour chaque opération il doit saisir les informations essentielles avant de valider l'opération sélectionné.

Une fois que l'expert valide l'opération, le système vérifie l'existence d'un fait portant les informations saisies par l'expert. C. à. d Il va passer par un test si le fait spécifié par le numéro entré déjà existe, et il va générer des exceptions en cas de défaillance par exemple :

Ajouter un fait qui déjà existe ou supprimer/renommer un fait qui n'existe pas, La [Figure2.5] et la [Figure2.6] illustre cela.

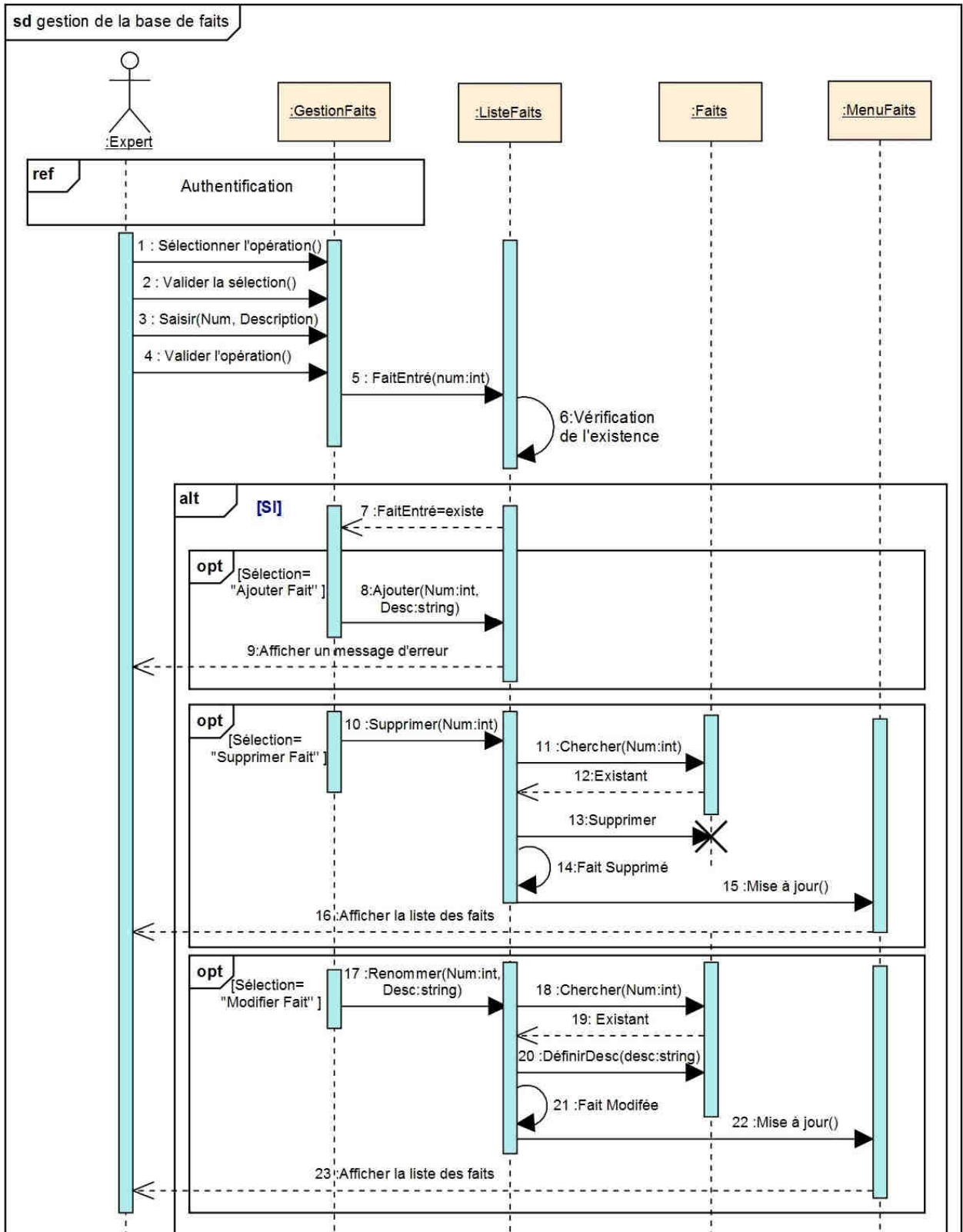


Figure2.5 : Diagramme de séquence de cas d'utilisation (gestion de base de faits 1).

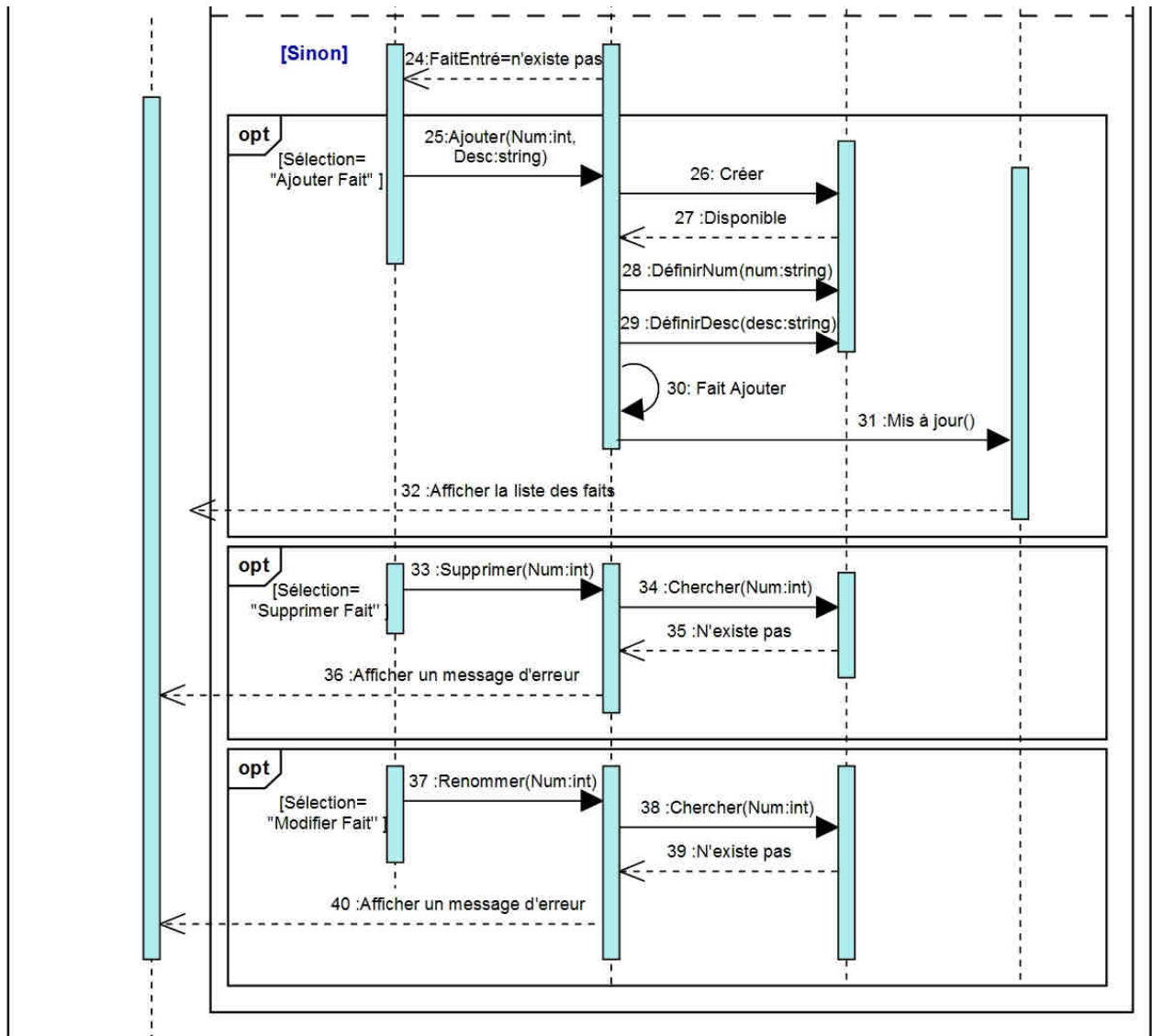


Figure2.6 : Diagramme de séquence de cas d'utilisation (gestion de base de faits 2).

1.3. Diagramme de séquence de cas d'utilisation (gestion de la base de règles):

Dans ce cas, l'expert entre dans sa session afin de gérer la base de règles, Où il sélectionne d'abord l'opération (soit l'ajout ou la suppression), et pour chaque opération il doit saisir les informations essentielles avant de valider l'opération sélectionné.

Une fois que l'expert valide l'opération, le système vérifier l'existence d'une règle portant les informations saisies par l'expert. C.à.d. Il va passer par un test si la règle spécifié par le numéro entré déjà existe, et il va générer des exceptions en cas de défaillance par exemple :

Ajouter une règle qui déjà existe ou supprimer une règle qui n'existe pas, La [Figure2.7] et la [Figure2.8] illustre cela.

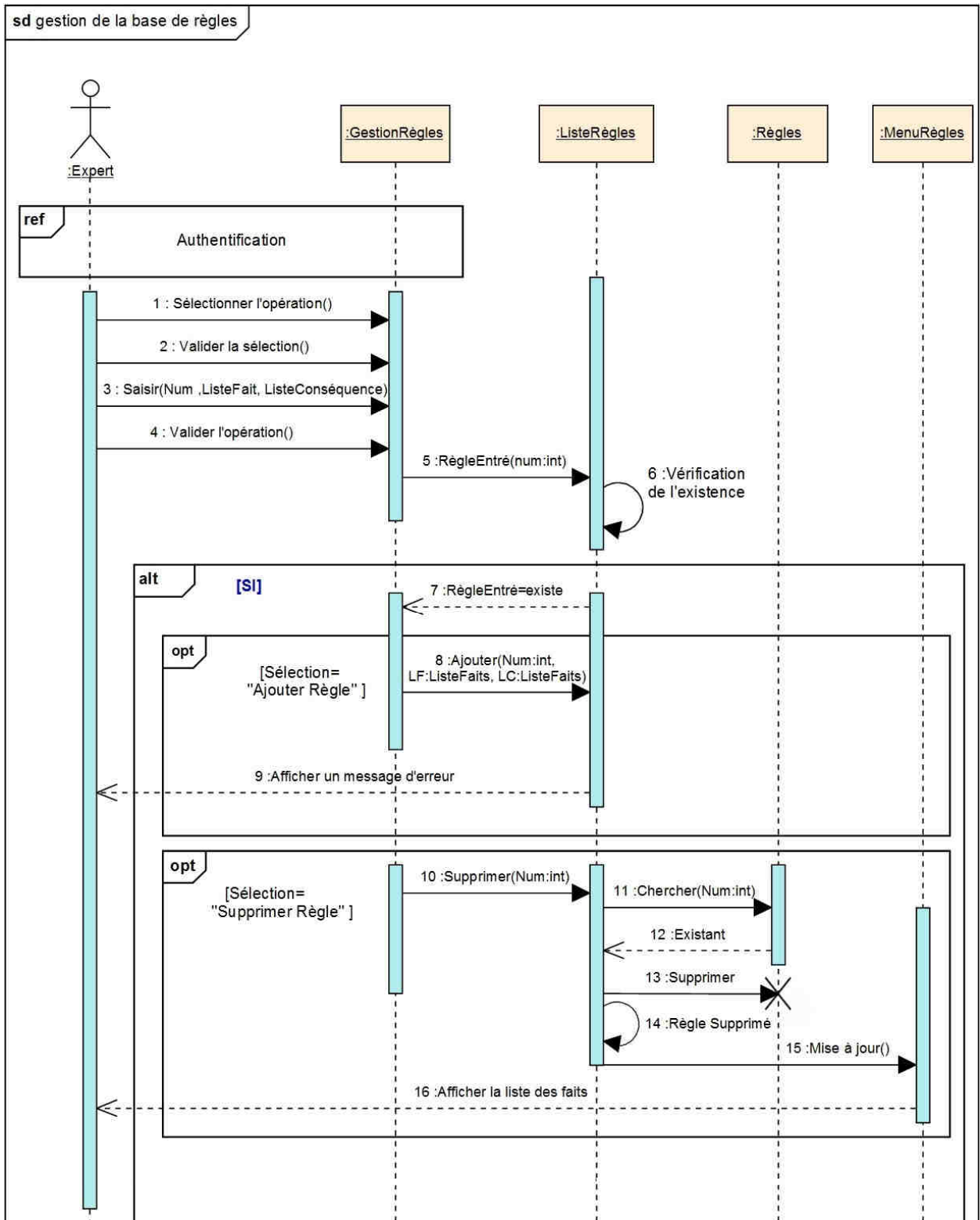


Figure2.7 : Diagramme de séquence de cas d'utilisation (gestion de base de règles 1).

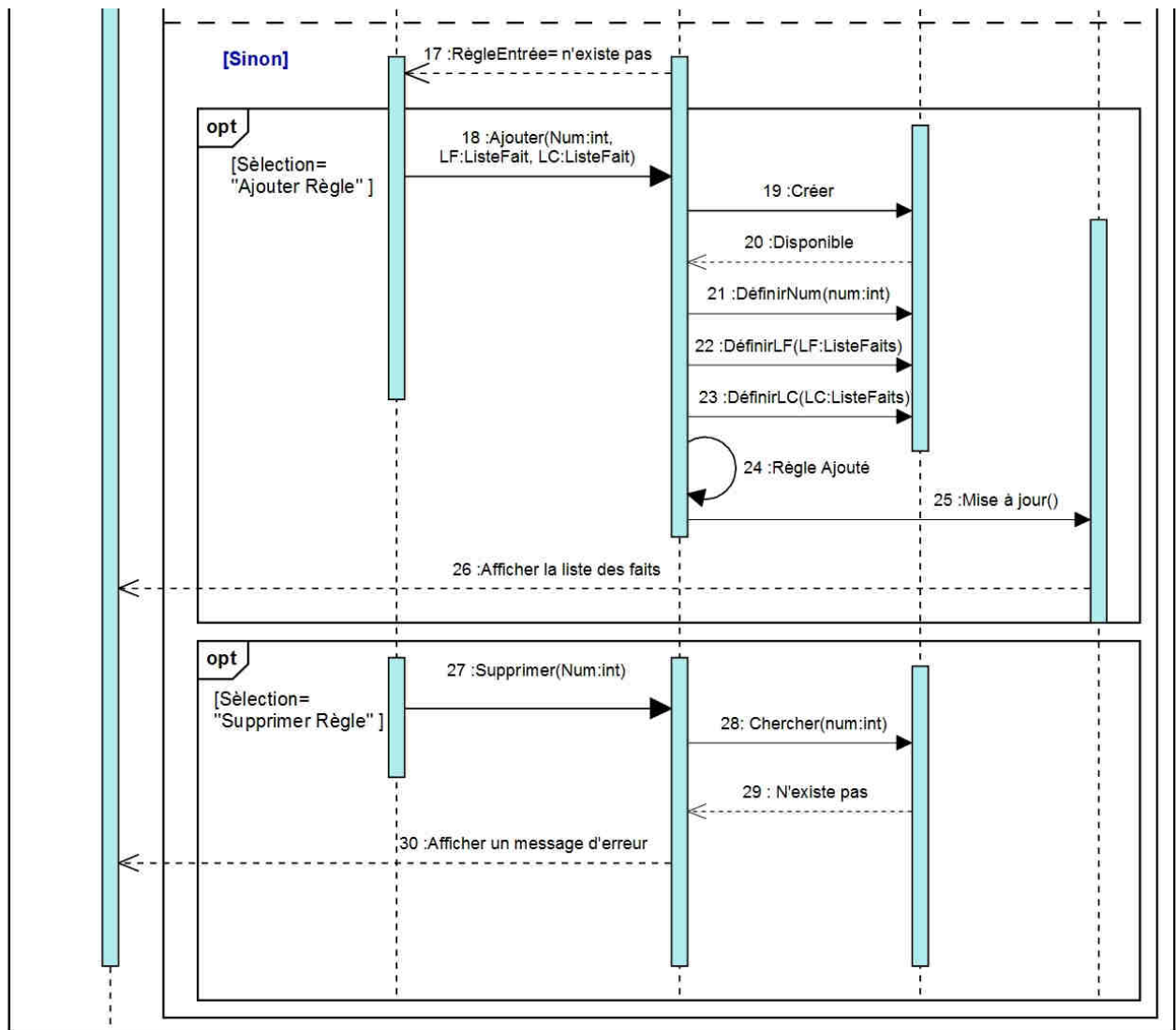


Figure2.8 : Diagramme de séquence de cas d'utilisation (gestion de base de règles 2).

1.4. Diagramme de séquence de cas d'utilisation (gestion de système expert):

Dans le cas de gérer un système expert, une fois que l'expert lance l'application un nouveau projet est créé automatiquement en début. En d'autres termes, tous les objets système sont créés automatiquement: liste de faits, liste de règle et tous les composants qui contiennent ces deux listes, et pour créer un nouveau projet, l'expert doit répéter le même processus.

Et après il doit gérer les deux bases : base de faits et base de règles. Voir les figures la [Figure2.5] et la [Figure2.6], [Figure2.7] et la [Figure2.8] précédente.

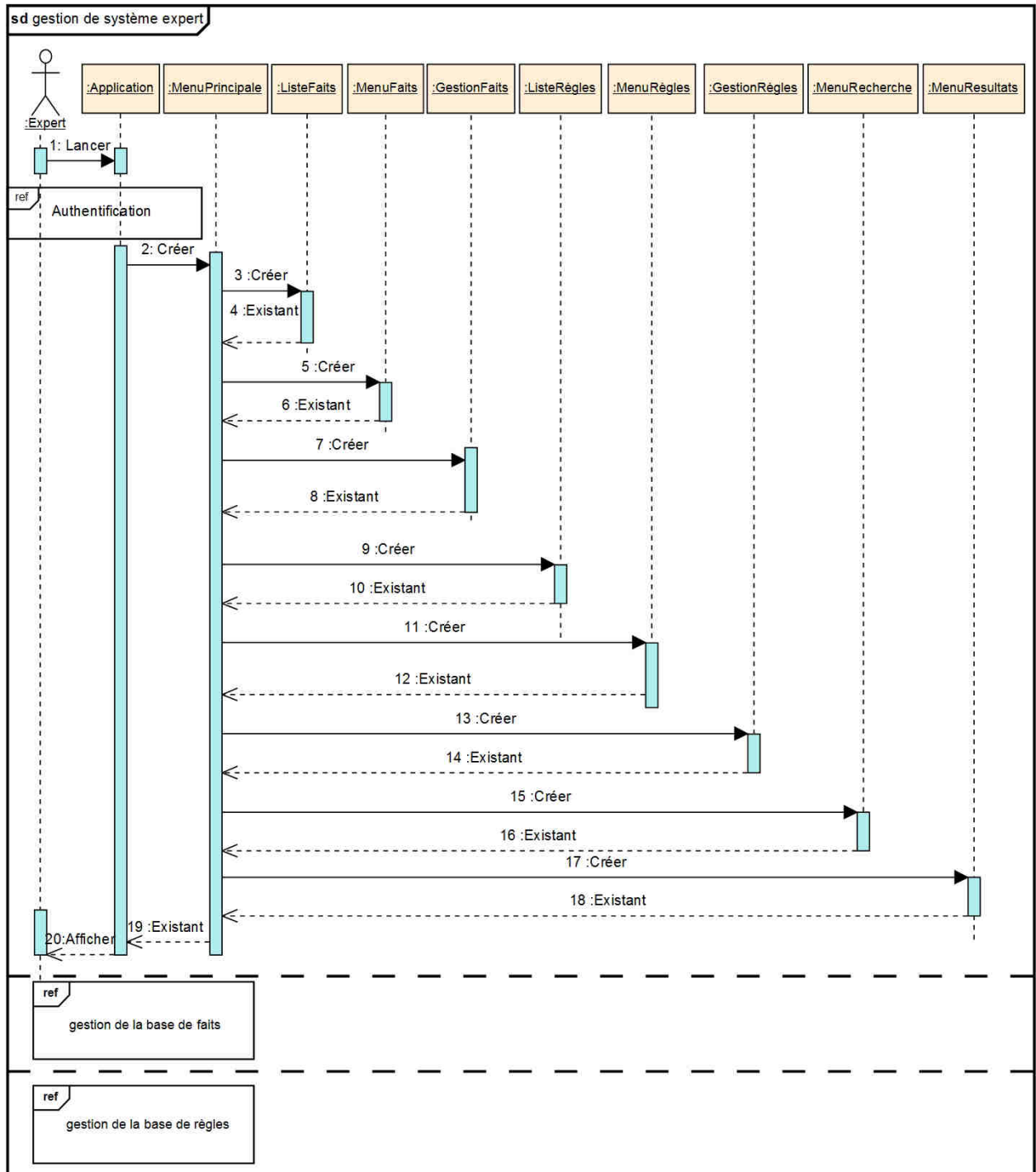


Figure2.9 : Diagramme de séquence de cas d'utilisation (gestion de système expert).

1.5. Diagramme de séquence de cas d'utilisation (Faire des recherches et inférences):

Puisque l'expert est l'administrateur de système, alors il peut faire des recherches dans le menu de recherche(MI) pour trouver des solutions sur ses problèmes, Où il sélectionne d'abord l'opération (soit chainage avant ou arrière), et pour chaque opération il doit saisir les faits initiaux (les symptômes) avant de valider l'opération sélectionné. Et après le moteur d'inférence va commencer son traitement et infère le résultat, Où il sauvegarde cette dernier dans une matrice. Ensuite, il le montre à l'expert.

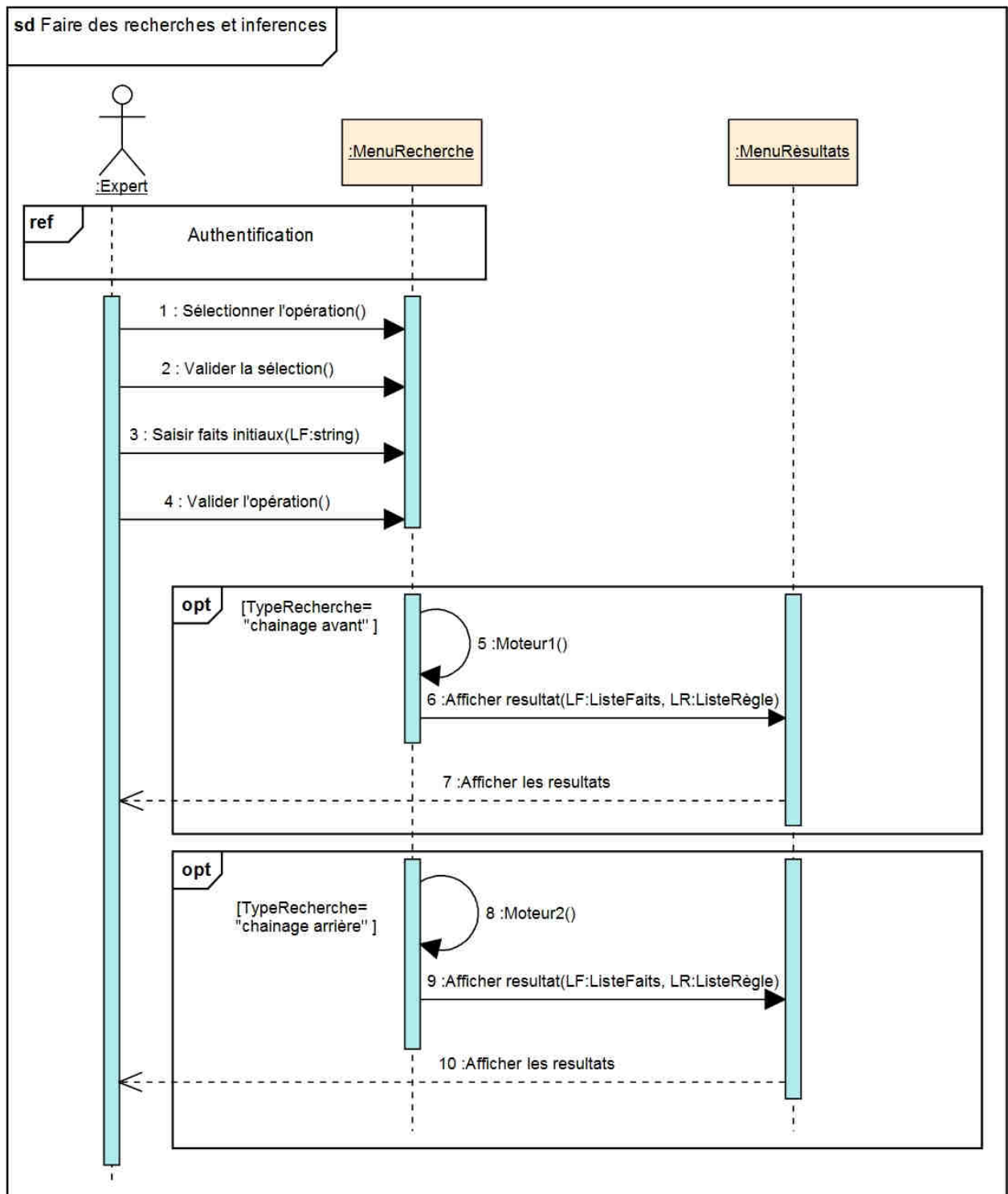


Figure2.10 : Diagramme de séquences de cas d'utilisation-Expert (Faire des recherches et inférences).

1.6. Diagramme de séquence de cas d'utilisation (Charger un projet):

Pour que L'expert puisse charger un projet, il doit en premier charger son base de faits et son base de règles, et après le système affiche les deux bases à l'expert.

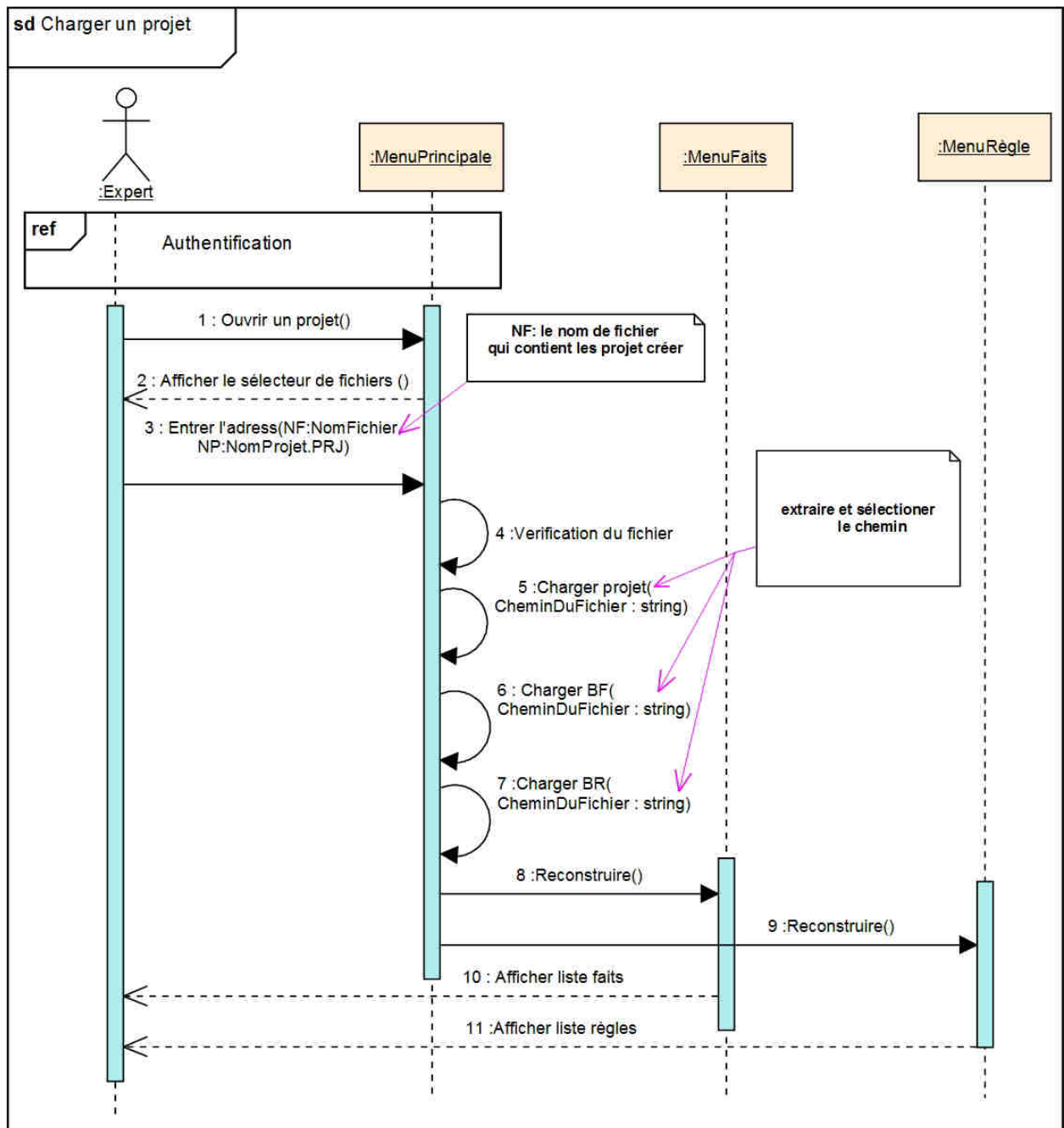


Figure2.11 : Diagramme de séquences de cas d'utilisation-Expert (Charger un projet).

1.7. Diagramme de séquence de cas d'utilisation (Sauvegarder un projet):

Pour que L'expert puisse sauvegarder un projet, il doit en premier sauvegarder son base de faits et son base de règles, pour s'assurer que le projet est enregistré avec succès.

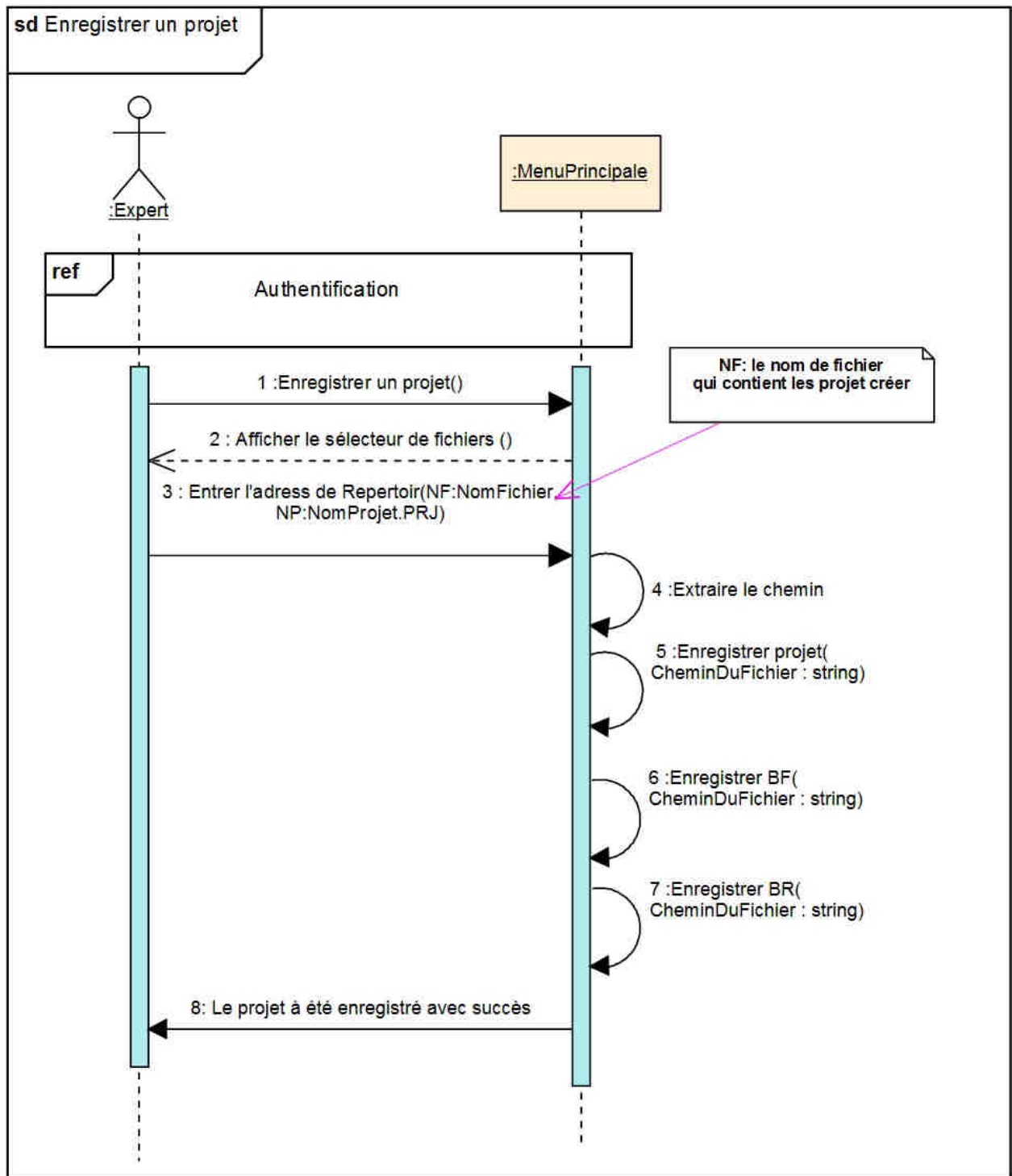


Figure2.12 : Diagramme de séquences de cas d'utilisation (Sauvegarder un projet).

- **Coté « Utilisateur »**

1.8.Diagramme de séquence de cas d'utilisation (Charger un projet):

L'utilisateur peut également télécharger un projet depuis son espace, où il télécharge d'abord son base de faits et son base de règles, et après le système affiche les deux bases à l'utilisateur.

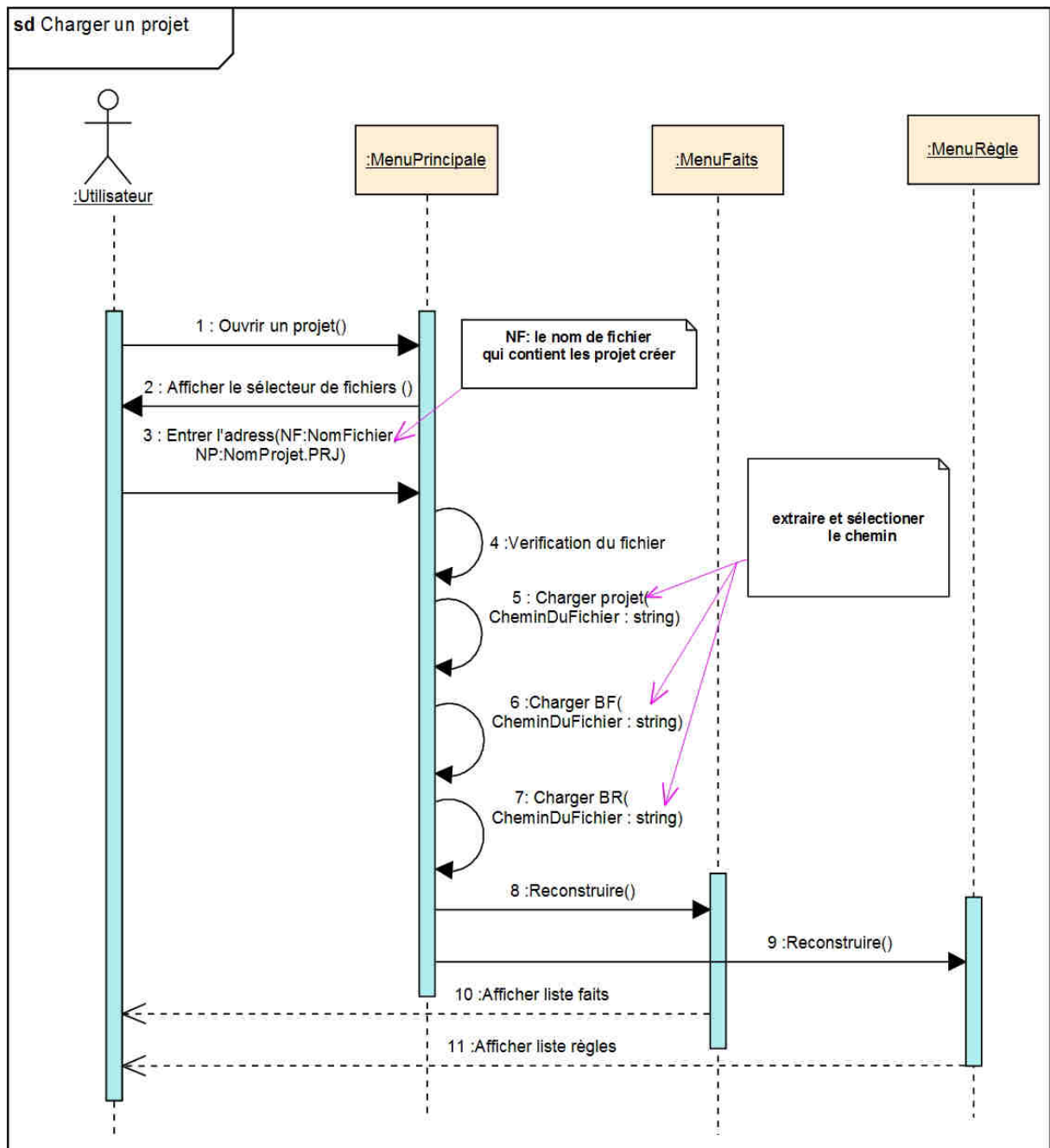


Figure 2.13 : Diagramme de séquences de cas d'utilisation-Utilisateur (Charger un projet).

1.9. Diagramme de séquence de cas d'utilisation (Faire des recherches et inférences):

Pour que L'utilisateur peut lancer une recherche sur des solutions à ses problèmes, dans le menu de recherche (moteur d'inférence), il doit spécifier la nature de la recherche (soit chaînage avant ou arrière), puis il doit spécifier une liste des faits initiaux a fin que le moteur d'inférence va commencer son traitement et infère le résultat, Où il sauvegarde cette dernier dans une matrice. Ensuite, il le montre à l'utilisateur.

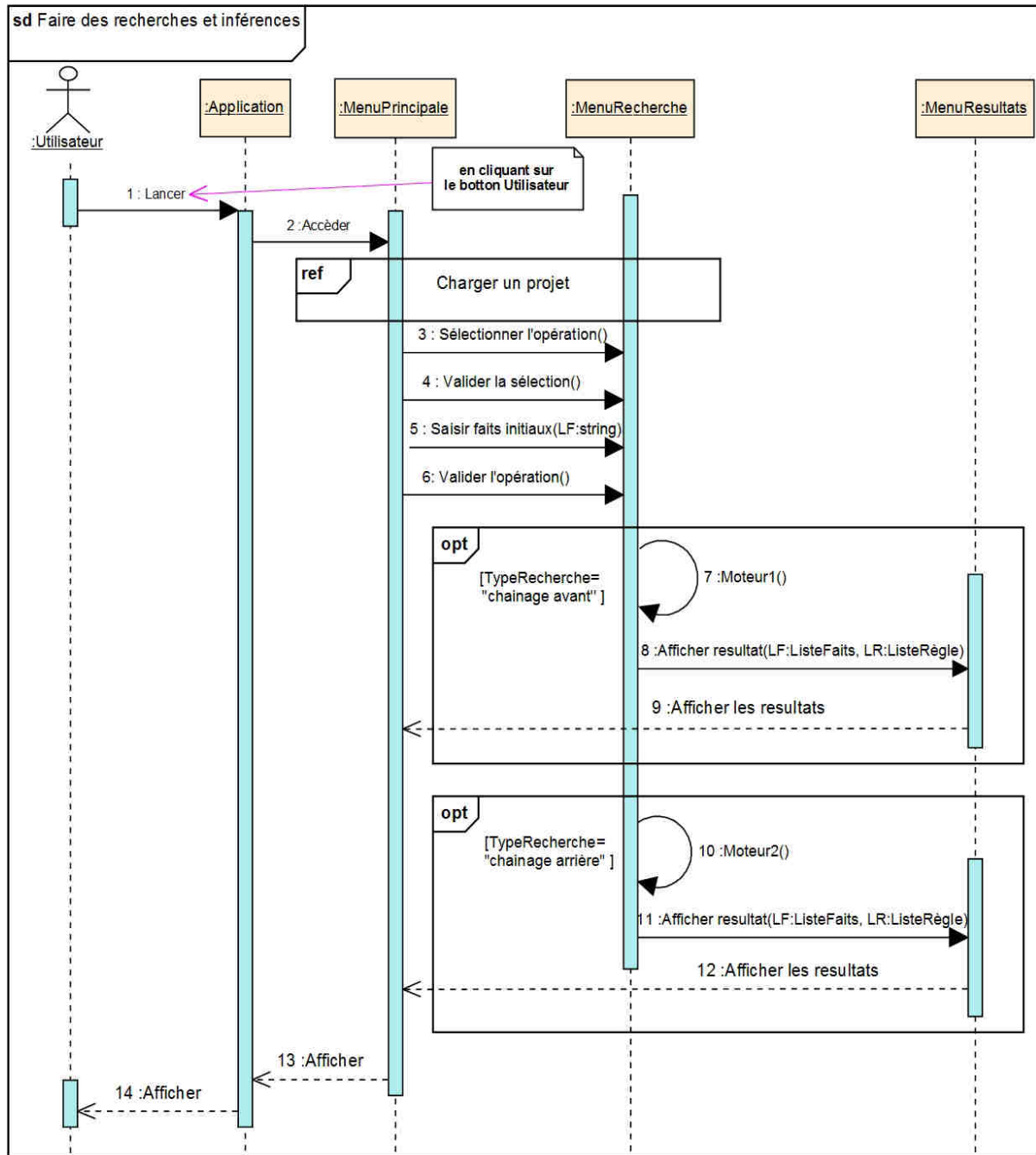


Figure 2.14 : Diagramme de séquences de cas d'utilisation-Utilisateur (Faire des recherches et inférences).

C. Diagramme de cas classe :

❖ Définition

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres

classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classe ; seul l'algorithme général et le pseudo-code correspondant peuvent être associés à la modélisation. [22].

❖ Description du diagramme de classe :

Le diagramme de classes identifie les classes nécessaires de notre système et les associations entre elles, Où il est composé de plusieurs menus, chaque menu est contenu deux composant essentiel sont la liste des faits, et la List des règles, en plus de cela notre système interagit avec deux acteurs (deux classes) principale sont : l'expert et l'utilisateur, où chacun joue son rôle.

✚ **Les règles de gestions (pour l'expert) :**

- Un expert a un seul compte (il est caractérisé par un nom d'utilisateur et un mot de passe).
- Un expert peut gère plusieurs bases de connaissances (base de faits, base des règles).
- Un expert peut faire plusieurs recherches.
- Un expert peut consulter plusieurs résultats.

✚ **Les règles de gestions (pour l'utilisateur) :**

- Chaque utilisateur peut faire plusieurs recherches.
- Chaque utilisateur peut consulter plusieurs résultats.

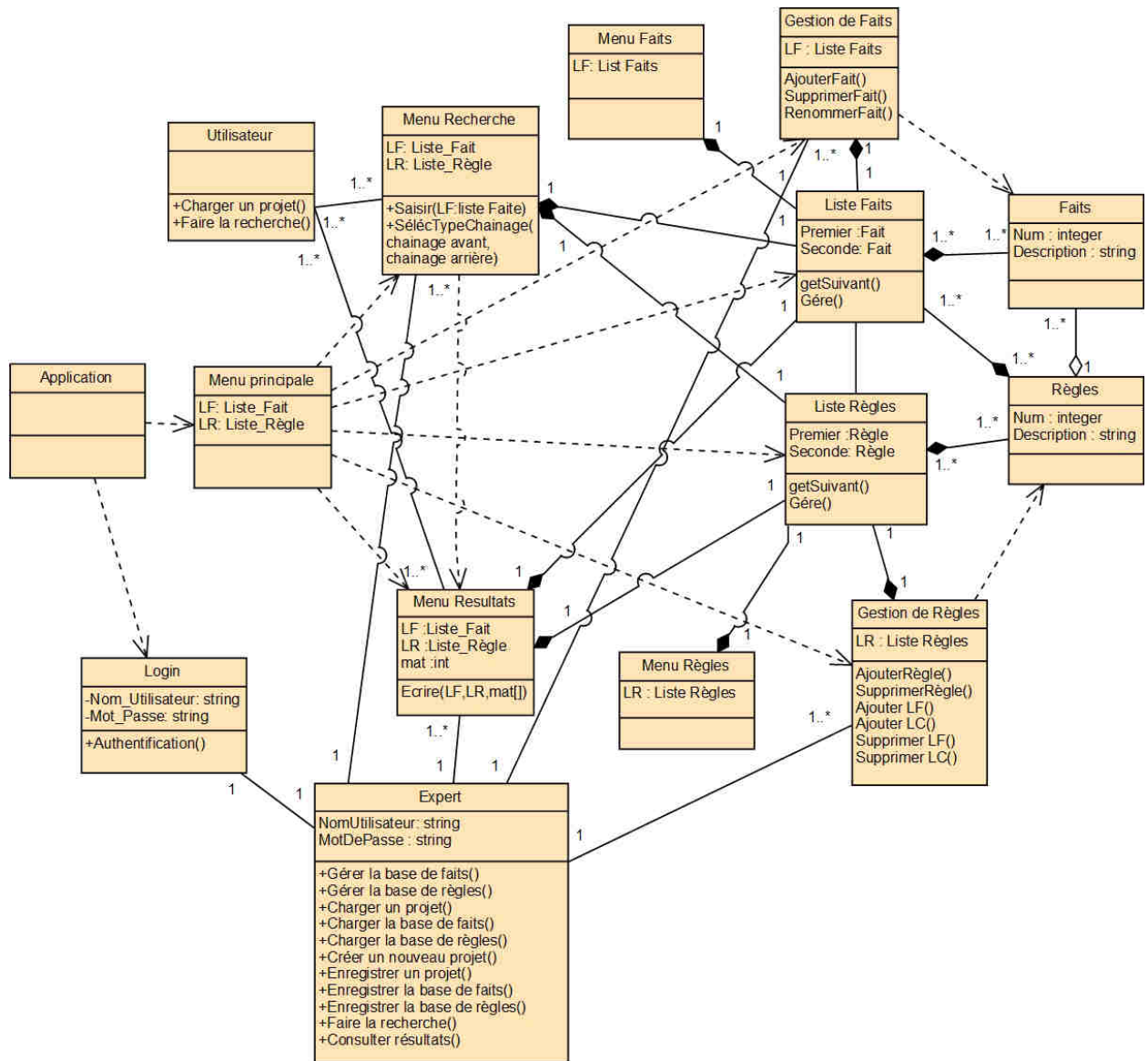


Figure2.15 : Diagramme de classe.

4. Comparaison entre les travaux existant

À travers les études que nous avons menées sur ce sujet dans le domaine informatique, nous avons trouvé des différences dans la méthode de réalisation de certains systèmes experts, où chacun diffère de l'autre avec un certain avantage. Donc ce tableau représente la comparaison entre les différents systèmes experts.

Remarque :

- **COMBREDES** : Conception et réalisation d'un Système Expert d'aide au diagnostic de pannes d'un Microordinateur (université de Bejaia - Alger).
- **CISEADPO** : Conception et implémentation d'un système expert d'aide au diagnostic de pannes d'un PC (université de Brouira – Alger).

Le nom de SE	SEDIP (notre système)	COMBREDES	CISEADPO
La description	<p>1- un système expert dynamique :</p> <p>- l'expert peut créer plusieurs projets dans tout les domaines, pas seulement dans le domaine de l'informatique.</p> <p>2-utilisation du base de connaissance, parce que il est plus fort que la base de donne.</p> <p>3-SEDIP est un système intelligent : fait l'apprentissage pour trouver des résultats logiques.</p> <p>4 - Le résultat n'est pas limité : le système affiche plusieurs résultats, avec ses probabilités.</p> <p>5-le SEDIP utilise le chainage avant et arrière.</p>	<p>1- un système expert non dynamique : - l'expert crée un formulaire qui contient des questions qui sont statique,</p> <p>- limité en horizon si on spécifie les options des questions.</p> <p>2-utilisation d'une base de donne.</p> <p>3- COMBREDES : n'est qu'un système de recherche.</p> <p>4-le résultat est limité : le système affiche un seul résultat à chaque opération.</p>	<p>1-un système expert non dynamique : -l'expert ne peut créer qu'un seul projet dans un domaine spécifique.</p> <p>2-utilisation d'une base de donne.</p> <p>3-CISEADPO n'est qu'un système de recherche.</p> <p>4-le résultat est limité : le système affiche un seul résultat à chaque opération.</p> <p>5-le CISEADPO utilise le chainage avant</p>

Tableau2.3 : Comparaison entre les systèmes experts.

5. Conclusion

Dans ce chapitre, nous nous sommes concentrés sur la conception de notre système, où nous avons mettons une architecture complète avec des étapes essentiels, et aussi nous avons également adopté la méthode UML pour modéliser notre système à travers les différents diagrammes (diagramme de cas d'utilisation, diagramme de séquence, diagramme de classe), et pour la assurer un bon fonctionnement nous nous accomplirons la réalisation de notre système dans le chapitre qui suite.

Chapitre 3 : Réalisation et Test

1. Introduction

La phase de réalisation est considérée comme l'étape la plus importante dans le cycle de développement de tout projet informatique, et Nous devons nous assurer qu'il répond vraiment aux besoins attendus. Donc L'objectif de ce chapitre est de présenter les outils, et les langages utilisés pour la mise en œuvre de notre prototype, et en plus, de citer l'organigramme de l'application, et après de décrire l'aspect implémentation, présenter l'application et les tests réalisés.

2. les outils et les langages utilisés

2.1. Les outils utilisés

A. Eclipse (l'environnement de développement)

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions [W12].



Figure3.1 : Logo Eclipse.

- La spécificité d'Eclipse IDE (IntegratedDevelopmentEnvironement) vient du fait de son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

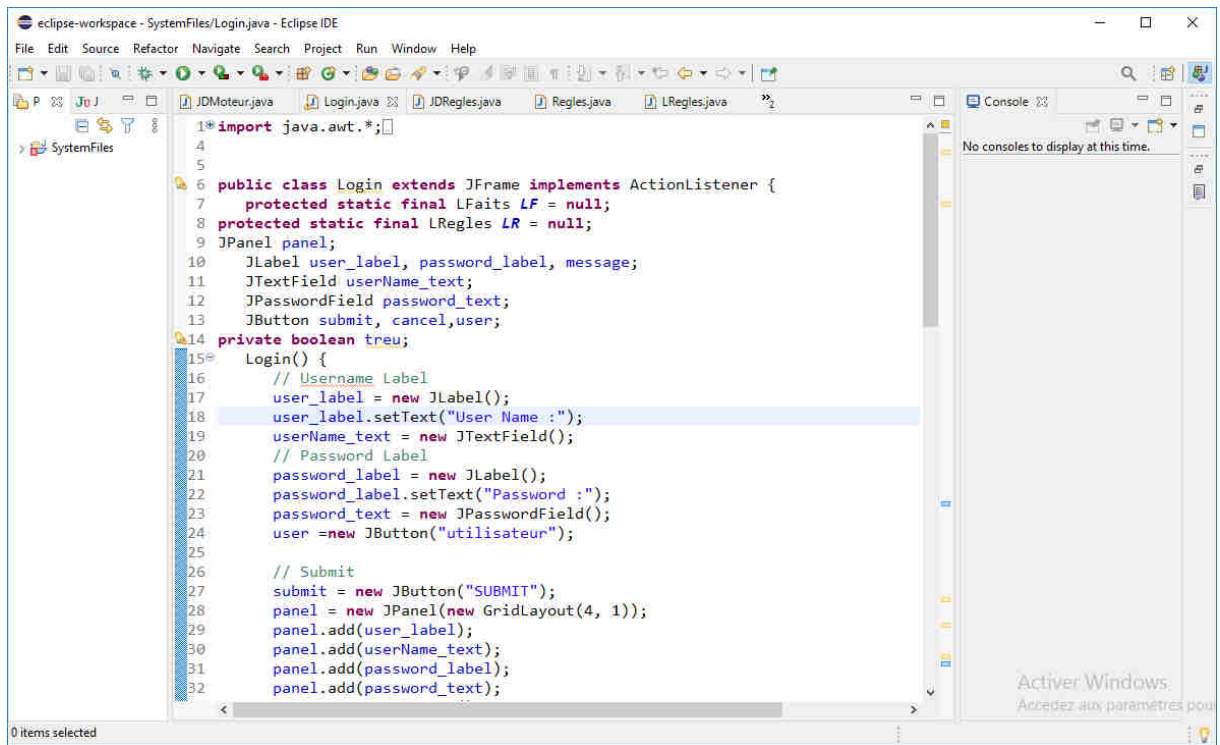


Figure3.2 : Eclipse IDE.

B. Java JDK

Le Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peuvent être compilé, transformé en bytecode destiné à la machine virtuelle Java. [W13].



Figure3.3 : Logo java JDK.

C. StarUML

StarUml est un logiciel de modélisation UML, qui a été "cédé comme open source " par son éditeur, à la fin de son exploitation commerciale (qui visiblement continue ...), sous une licence modifiée de GNU GPL. Aujourd'hui la version StarUML V3 n'existe qu'en licence propriétaire.

StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0. StarUML est écrit en Delphi, et dépend de composants Delphi propriétaires (non open-source). [W14].



Figure3.4 : Logo StarUML.

D. Pacestar

Pacestar dispose d'un module WYSIWYG, supporte l'OMG UML 2.0, des fonctions classiques de diagramme, il supporte en import et export les fichiers JPG,GIF,GIF,BMP,DIB,WMF. Cet outil permet d'insérer des liens HyperText vers d'autres diagrammes et fichiers externe. Ce logiciel est basé sur UML 2.0 défini par OMG. [W15].



Figure3.5 : Logo Pacestar.

2.2. Les langage utilisés

A. Java

Le langage Java est un langage généraliste de programmation synthétisant les principaux langages existants lors de sa création en 1995 par Sun Microsystems. Il permet une programmation orientée-objet (à l'instar de SmallTalk et, dans une moindre mesure, C++), modulaire (langage ADA) et reprend une syntaxe très proche de celle du langage C. Outre son orientation objet, le langage Java a l'avantage d'être modulaire (on peut écrire des portions de code génériques, c.-à-d. utilisables par plusieurs applications), rigoureux (la plupart des erreurs se produisent à la compilation et non à l'exécution) et portable (un même programme compilé

peut s'exécuter sur différents environnements). En contrepartie, les applications Java ont le défaut d'être plus lentes à l'exécution que des applications programmées en C par exemple [W16].



Figure3.6 : Logo Java.

3. L'organigramme de notre application

Notre système contient deux espaces, un espace pour l'utilisateur et l'autre pour l'expert, chacun jouant son rôle dans son espace, et ce dernier est constitué des menus et sous menus.

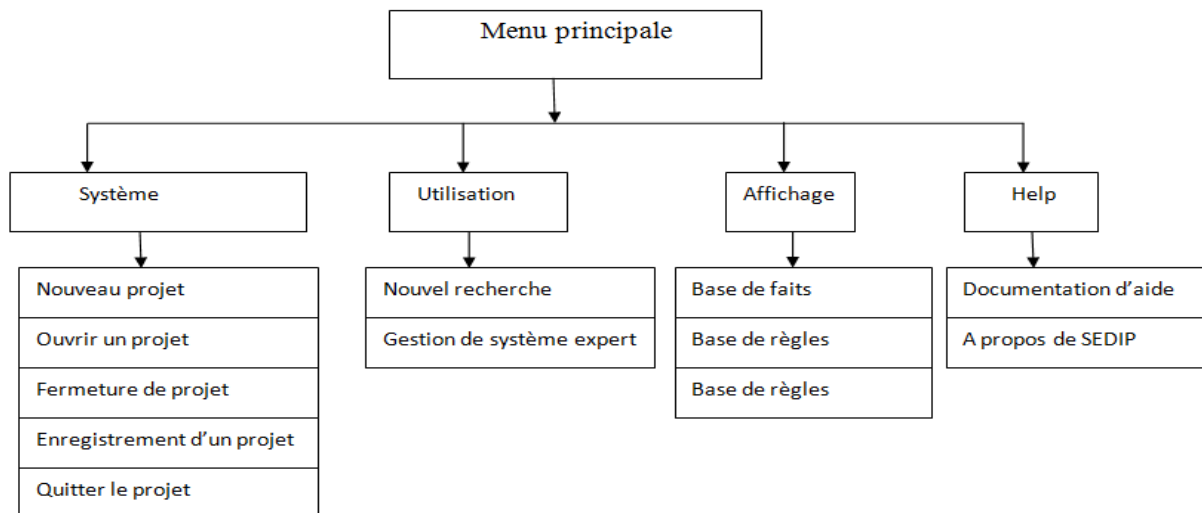


Figure3.7 : Organigramme de l'application.

4. Présentation et test de l'application

Dans cette partie, nous allons présenter quelques interfaces de notre système (SEDIP).

4.1. Page d'authentification

Grâce à cette page, l'expert peut se connecter à sa session où il saisit ses informations, qui sont le nom d'utilisateur et le mot de passe et après, il clique sur le bouton connexion.

Quant à l'utilisateur, il n'appuie que sur le bouton utilisateur.



Figure3.8 : Page d'authentification.

4.2. Menu principale

Notre application contient deux menus principaux, chaque menu représente un espace privé pour un acteur, le premier pour l'expert et le deuxièmement pour l'utilisateur.

4.2.1. Espace expert

L'espace expert contient de nombreux menus, qui peuvent être affichés via les boutons de la barre d'outils ci-dessus, avec l'aide de ce dernier, l'expert peut gérer un système expert (gérer BDF, gérer BDR), et il peut également faire des recherches et créer de nouveaux projets, puis il peut stocker des données pour chaque projet ou télécharger Base de données ou projet.



Figure3.9: Espace Expert.

4.2.1.1. Menu de gestion de la base des faits

Le menu de gestion de la base des faits englobe deux menus :

- Menu pour faire la gestion des faits : Dans ce menu, l'expert gère la base de faits, C.-à-d. il fait des opérations sur les faits comme l'ajout, la suppression, renommer.
- Menu pour l'affichage des faits de la base : grâce à ce menu, l'expert peut connaître le nombre des faits dans la base.

La [Figure3.10] Ci-dessous montre le premier menu dans le cas où la liste de faits est vide, Et la [Figure3.11] montre le deuxième menu dans le cas où la liste de faits est rempli.

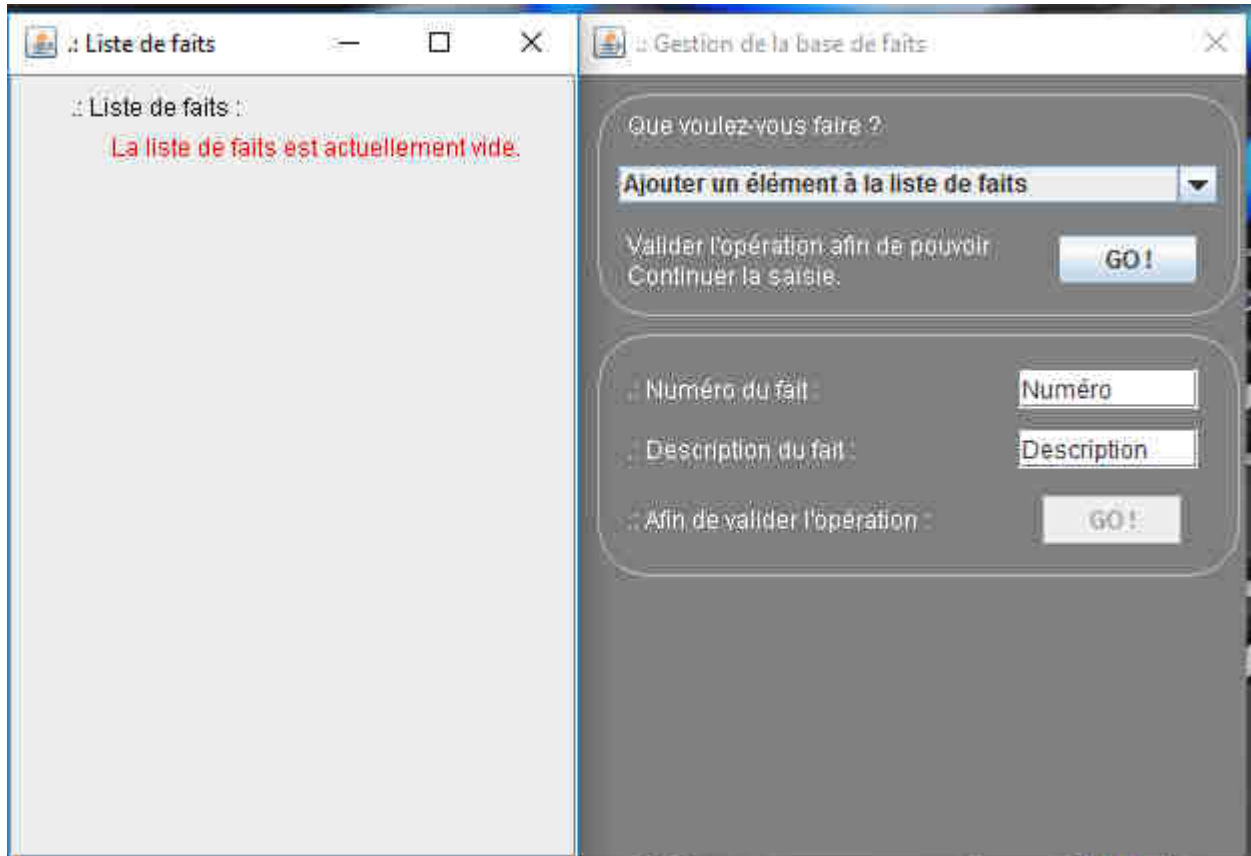


Figure3.10 : Gestion des faits (vide).

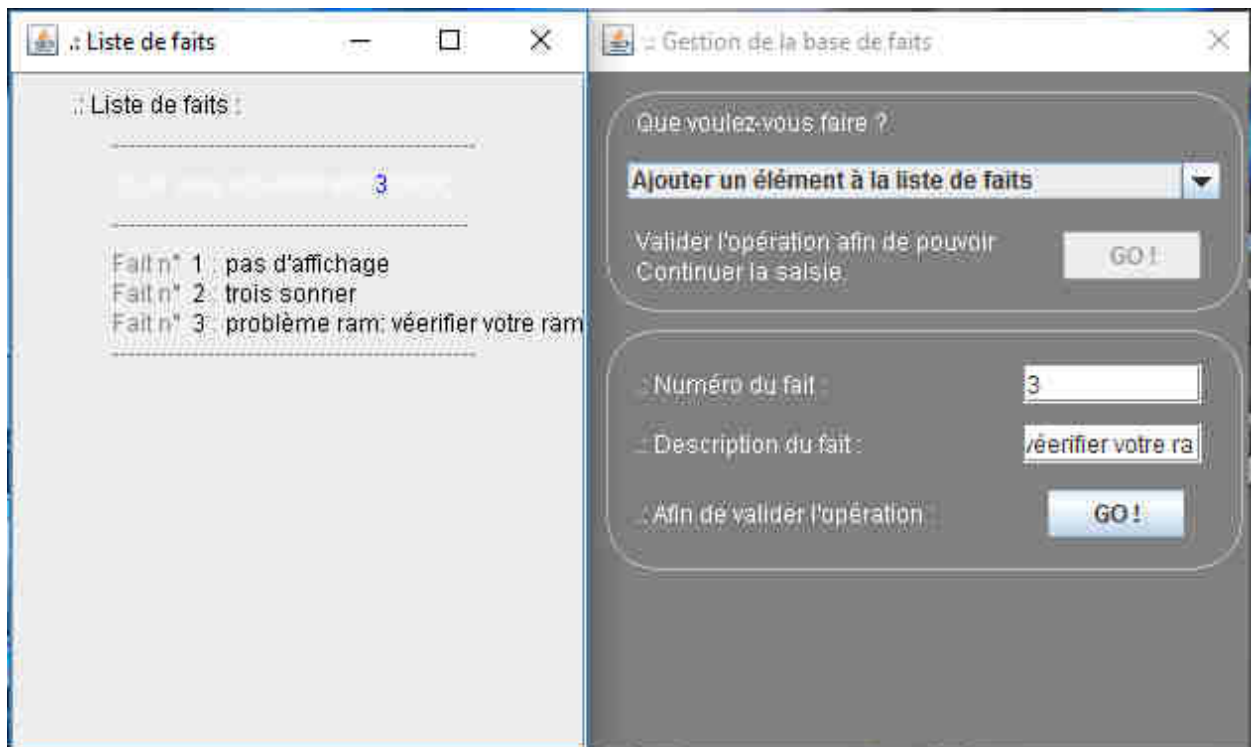


Figure3.11 : Gestion des faits (rempli).

4.2.1.2. Menu de gestion de la base des règles

Le menu de gestion de la base des règles englobe deux menus :

- Menu pour faire la gestion des règles : Dans ce menu, l'expert gère la base de règles, C.-à-d. il fait des opérations sur les règles comme l'ajout, la suppression.
- Menu pour l'affichage des règles de la base : grâce à ce menu, l'expert peut connaître le nombre des règles dans la base.

La [Figure3.12] Ci-dessous montre le premier menu dans le cas où la liste de règles est vide, et la [Figure3.13] montre le deuxième menu dans le cas où la liste de règles est rempli.

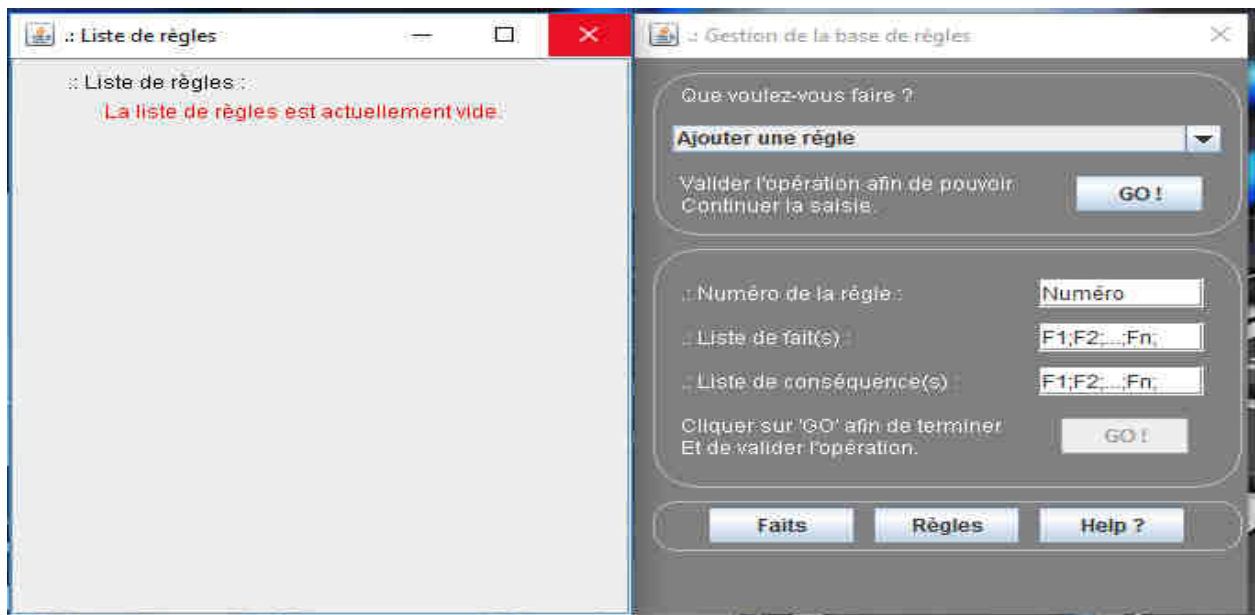


Figure3.12 : gestion des règles (vide).

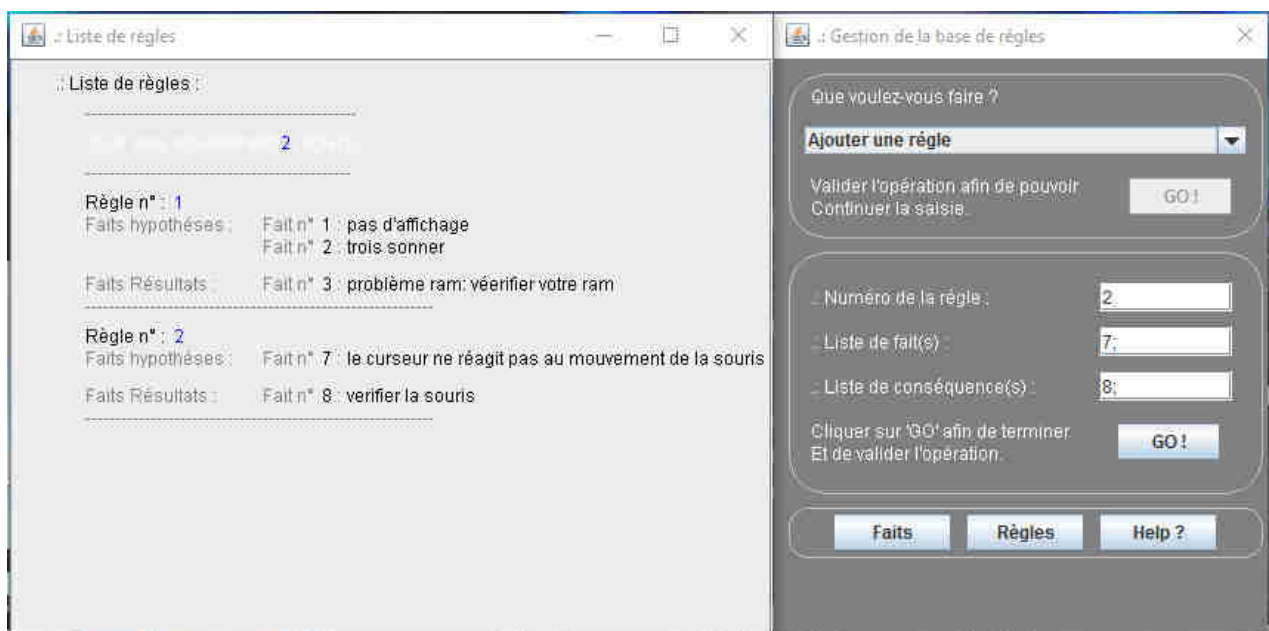


Figure3.13 : gestion des règles (rempli).

4.2.1.3. Menu de gestion des recherches et inférences

Dans ce menu, l'expert recherche des solutions à de nombreux problèmes, où ce dernier englobe deux menus:

- Menu pour faire la recherche : Dans ce menu, l'expert sélectionne le type de l'opération soit chainage avant ou chainage arrière, et après il saisit liste des faits initiaux.
- Menu pour l'affichage des résultats : grâce à ce menu, l'expert peut voir le résultat qui attends.
- Le SEDIP infère les résultats et lui afficher a l'expert.

La [Figure3.14] Ci-dessous montre le menu avant le lancement de recherche, C.-à-d. avant de sélectionner aucune opération, et avec un résultat vide.

La [Figure3.15] montre la première opération c.-à-d. le chainage avant avec ses résultats.

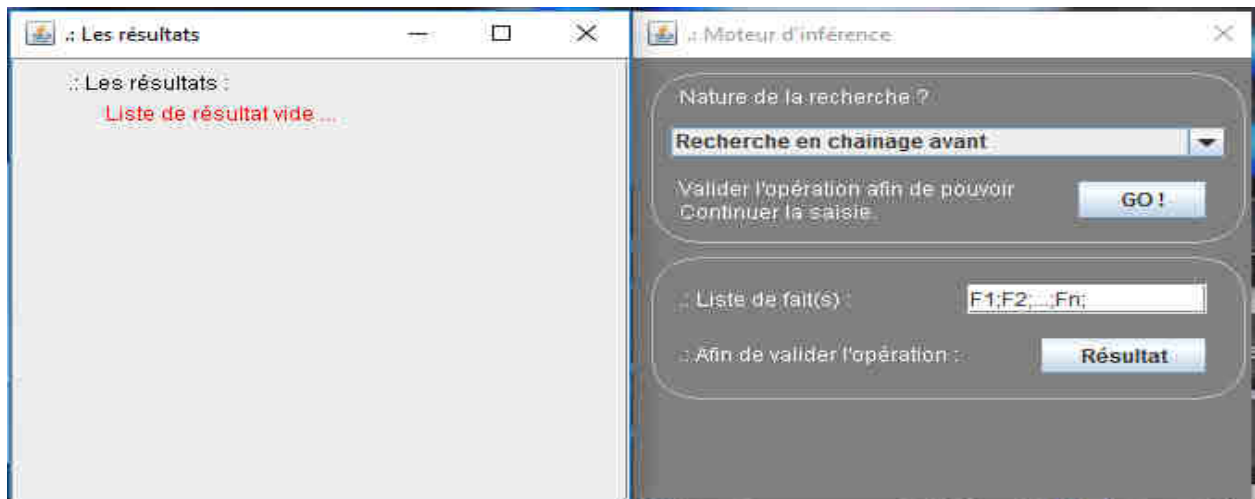


Figure3.14 : gestion de recherche

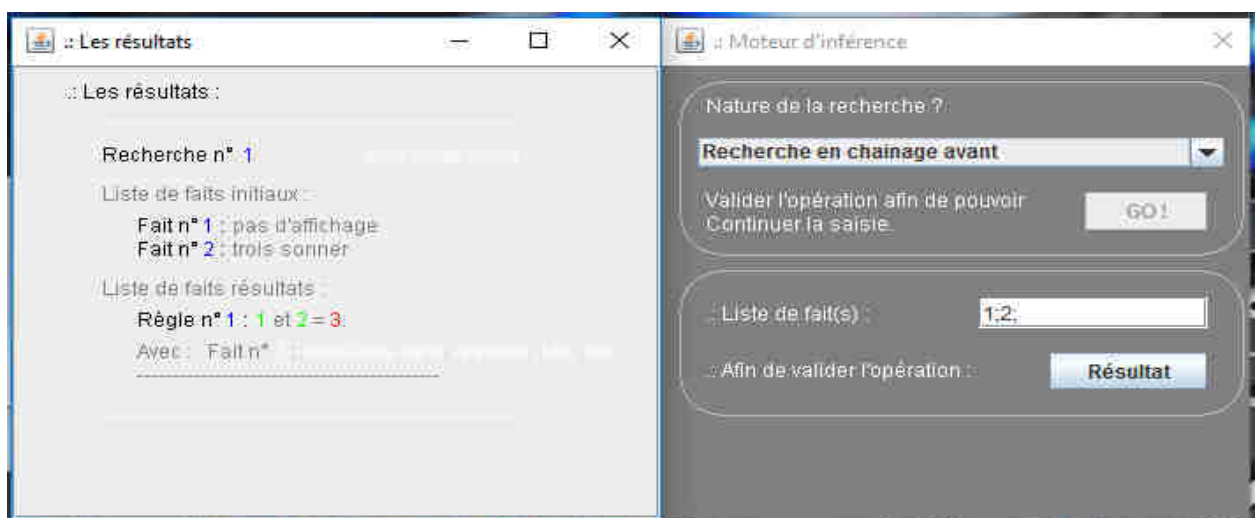


Figure3.15 : gestion de recherche avec chainage avant.

La [Figure3.16] montre la deuxième opération c.-à-d. le chainage arrière avec ses résultats.

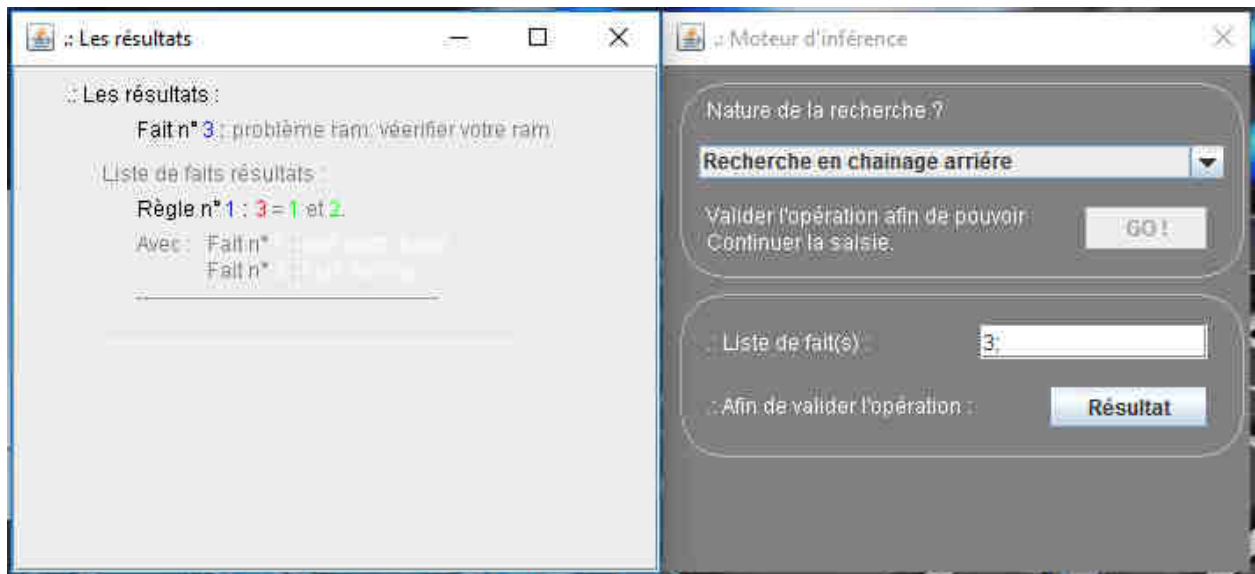


Figure3.16 : gestion de recherche avec chaînage arrière.

4.2.2. Espace utilisateur

L'espace utilisateur contient de nombreux menus, qui peuvent être affichés via les boutons de la barre d'outils ci-dessus, avec l'aide de ce dernier, l'utilisateur peut télécharger un projet ou une base de connaissances pour faire des nouvelles recherches, et aussi il peut consulter le fichier help pour savoir plus des informations.



Figure 3.17 : Espace Utilisateur.

4.2.2.1. Menu de gestion des recherches et inférences

Dans ce menu, l'utilisateur recherche des solutions à de nombreux problèmes, où ce dernier englobe deux menus:

- Menu pour faire la recherche : Dans ce menu, l'utilisateur sélectionne le type de l'opération soit chainage avant ou chainage arrière, et après il saisit liste des faits initiaux.
- Menu pour l'affichage des résultats : grâce à ce menu, l'utilisateur peut voir le résultat qui attends.
- Le SEDIP infère les résultats et lui afficher a l'utilisateur.

La [Figure3.18] montre la première opération c.-à-d. le chainage avant avec ses résultats.

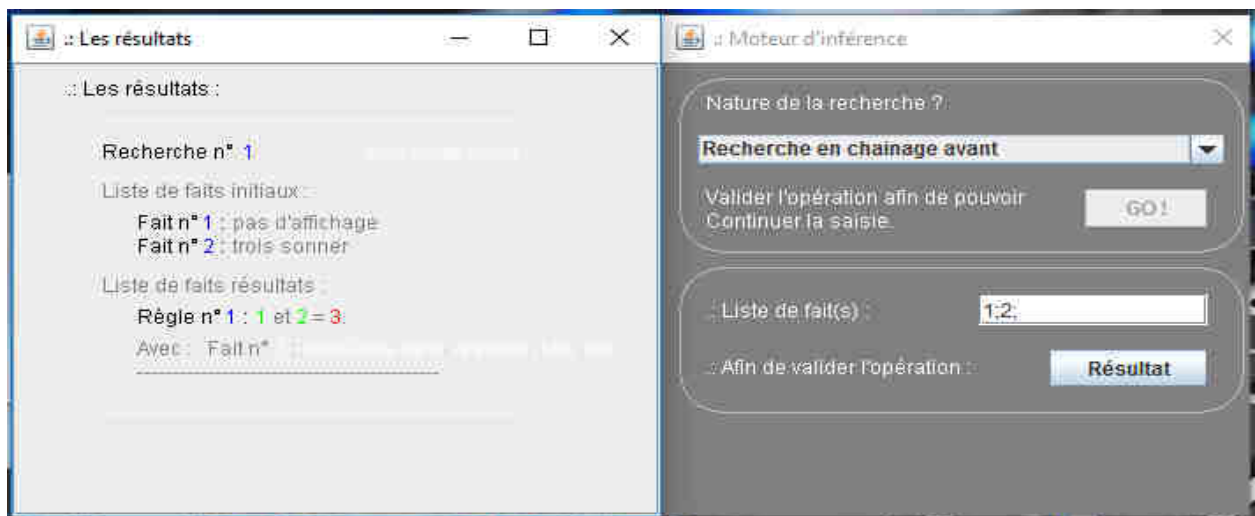


Figure3.18 : gestion de recherche avec chainage avant.

[Figure3.19] montre la deuxième opération c.-à-d. le chainage arrière avec ses résultats.

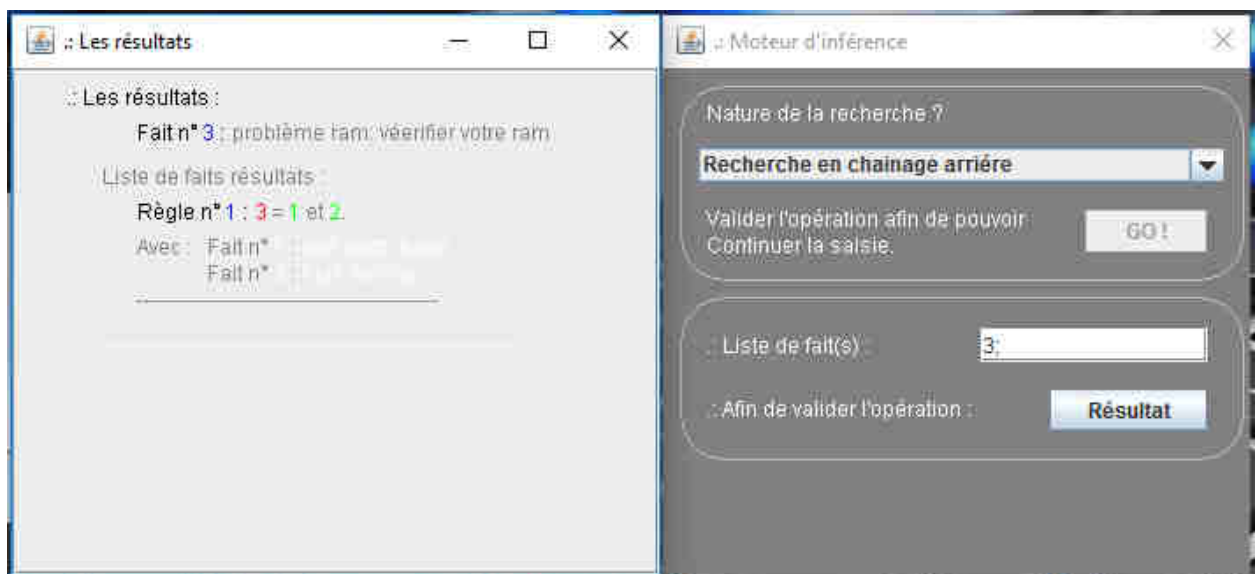


Figure3.19 : gestion de recherche avec chainage arrière.

4.2.2.2. Consulter le fichier help

Le fichier help facilite l'utilisation de l'application pour les utilisateurs simple. Il contient toutes les informations nécessaires pour faire des recherches, et pour comprendre les résultats qui afficher dans le menu de résultats avec des exemples. Voir la [Figure3.20].

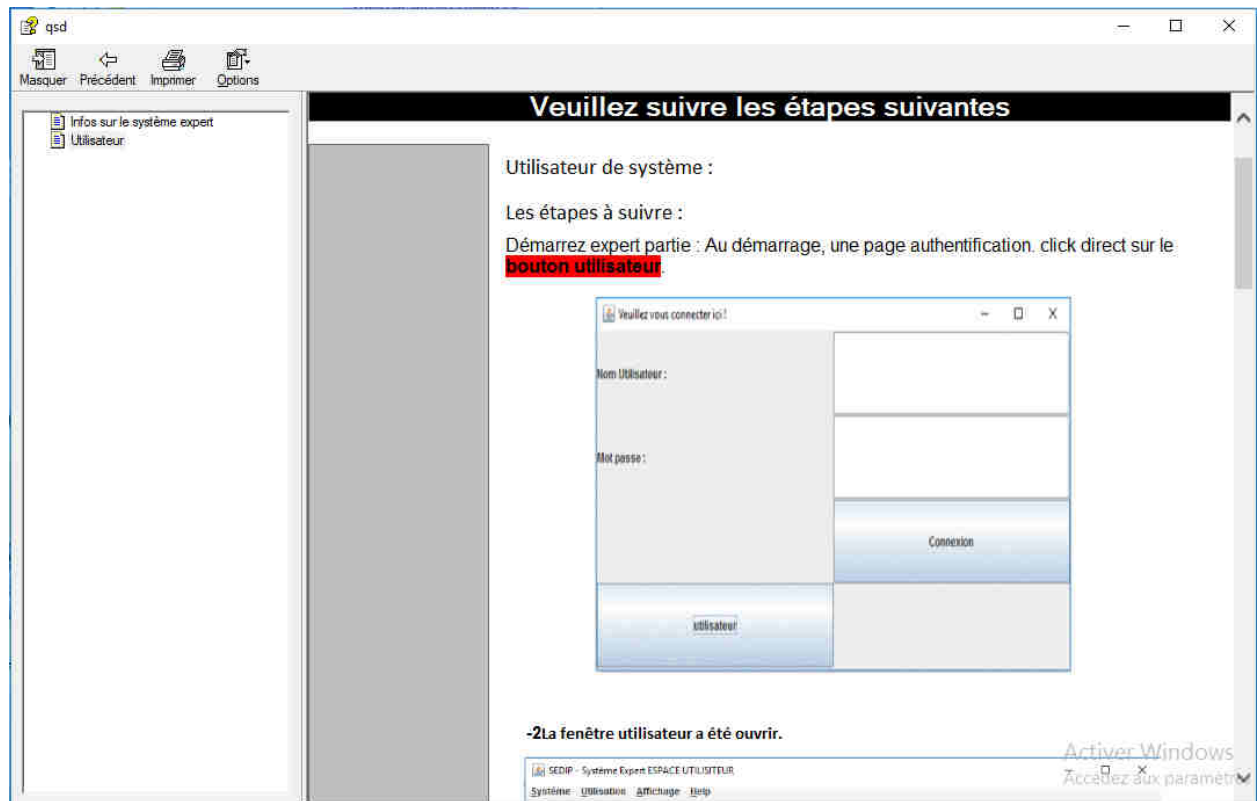


Figure3.20 : Fichier help.

4.3.A propos de SEDIP :

Ce fichier contient les informations nécessaires sur notre système SEDIP, qui comprend le nom de système, et le nom du développeur et de l'encadreur de ce projet.

La [Figure3.21] Ci-dessous montre quelques informations sur notre système.



Figure3.21: A propos de SEDIP.

5. Conclusion

Dans ce chapitre, nous avons réalisés un système expert capable de détecter les pannes d'un micro-ordinateur, en utilisant certains outils et langages, notamment le langage Java et l'environnement de développement "Eclipse", et les logiciels de modélisation UML, StarUML et PaceStar.

Conclusion et perspectives

Dans cette mémoire, nous avons appris à connaître l'une des techniques d'intelligence artificielle, c'est les systèmes experts qui sont avant tout des logiciels qui simulent le raisonnement d'un expert.

Au début de ce travail, nous avons d'abord évoqué le problème lié à notre sujet (la maintenance informatique) est de savoir comment réaliser et créer un système expert qui nous aide à diagnostiquer les types de pannes qu'un micro-ordinateur rencontré. Ensuite nous avons définis l'intelligence artificielle et les systèmes experts et quelque concept qui concerne les deux. Dans la deuxième partie nous avons mis en place une structure générale de notre système (SEDIP), puis d'utilisé la méthode UML pour le concevoir, qui est un langage adéquat a notre cas et un outil idéal pour la programmation orienté objet, qui nous a permet de modéliser toutes nos idées a travers les différents diagrammes : cas d'utilisation, séquence, classes ...etc.

Enfin, nous avons étudié quelques outils pour l'implémenter dans le troisième chapitre, parmi eux, l'outil de développement Eclipse IDE avec le langage java, et les outils de conception nous avons choisi StarUML et PaceStar.

En conclusion, nous avons réalisé un système qui est capable de détecté les panne d'un micro-ordinateur nommé par SEDIP. Ce dernier est un système expert dynamique d'ordre 0 (logique des propositions), un outil d'aide à la décision qui permet de diagnostiquer des défaillances d'un micro-ordinateur, et aussi il utilise deux modes de raisonnement (chainage avant et chainage arrière). En plus de cela, dans le menu de résultat, il affiche plusieurs résultats (avec des probabilités), C.-à-d. que le résultat prioritaire est celui qui apparaît en haut de la liste (la solution la plus proche du problème).

Dans des recherches futures, nous allons essayer d'améliorer l'efficacité de notre système, améliorer le temps d'exécution, et d'enrichir notre base de connaissance.

A. Références Bibliographiques

- [1] د محمد زكي. أ محمد خضر
المؤتمر التقني الرابع لنظم الاتصالات و المعلومات في الدول الإسلامية. الأنظمة الخبيرة في التطبيقات الصناعية.
- [2] **K.Kasabov.** Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. London, England: The MIT Press, 1998.
- [3] [BAR 81].A. BARR - FEIGENBAUM The handbook of Artificial Intelligence Pitman LONDRES 1981 Traduction française - Edition Eyrolles 1986
- [4] Laurent Audibert (LIPN - UMR CNRS 7030) Université Paris 13 – Laboratoire d’Informatique de Paris-Nord (LIPN) 4 novembre 2010
- [5] P. Jackson. Introduction to Expert Systems. Addition-wesley, 1986, p50.
- [6] Osheroff JA, Teigh JM, Middleton B, Steen EB, Wright A, Detmer DE. A Roadmap for National Action on Clinical Decision Support. JAMIA 2007;
- [7] Abidi SSR. Healthcare knowledge management: The art of the possible. In: Knowledge Management for Health Care Procedures. Berlin: Springer, 2008;
- [8] G.F. Luger and W.A. Stubblefield. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Addition-wesley, 1999.
- [9] François Denis Laboratoire d’Informatique fondamentale, Marseille Laurent Miclet ENSSAT-IRISA, Lannion «INTELLIGENCE ARTIFICIELLE : LES SYSTEMES EXPERTS ». - Juillet 2009
- [10] Marion Charbonnier, Les systèmes experts Etat de l'art et application possible aux SIG, 25 Avril 2008
- [11] F. H. RoTH, D. A. WATERMAN, D. B. LEMAT, "Building Expert Systems", Editions Addison-Wesley, U.S.A., 1983.
- [12] J.-L. Laurière. Représentation et utilisation des connaissances. Techniques et science informatique 1(1):25-42 & 1(2):109-133, 1982
- [13] François Lévy Représentation des connaissances et logique modale MICR Institut Galilée - 5 mars 2006
- [14] A. AJITH, "Rule-based Expert Systems HEURISTICS," ed: Wiley Online Library, 2005, pp. 909-919.
- [15] LES RESEAUX DE NEURONES ARTIFICIELS INTRODUCTION AU CONNEXIONNISME COURS, EXERCICES ET TRAVAUX PRATIQUES - Claude TOUZET Juillet 1992

- [16] Descles Jean-Pierre. Réseaux sémantiques : la nature logique et linguistique des relateurs. In: *Langages*, 22^e année, n°87, 1987. Sémantique et intelligence artificielle, sous la direction de François Rastier. pp. 55-78.
- [17] Albéric Martel - Romain Bouleis Université de Savoie. MPI-2 les systèmes experts – 2007
- [18] William.Siler et J.Buckley. Fuzzy expert systems and fuzzy reasoning. New Jersey : John Wiley & Sons, 2005.
- [19] Konar Amit. Artificial Intelligence and Soft Computing Behavioral and Cognitive Modeling of the Human Brain. Calcutta, India : CRC Press, 2000
- [20] Application des réseaux de neurones et des systèmes experts Ecole de technologie supérieur – Université du Québec – 2011
- [21] PASCAL ROQUES et FRANCK VALLÉE, UML2 en action, De l'analyse des besoins à la conception, 4^{ème} édition. 2007
- [22] Joseph Gabay et David Gabay, UML Analyse et conception Dunod, Paris, 2008 ISBN 978-2-10-053567-5.
- [23] RumBaugh J., Jacobson I,Booch G. «UML 2.0 guide de référence.»CampusPress, 2004.

B. Références Web (Techniques)

- [W1] <http://intelart.e-monsite.com/pages/ii-l-utilisation-de-l-ia/domaines-d-applications.html>
- [W2] <https://www.axiocode.com/avantages-inconvenients-intelligence-artificielle/>
- [W3] <https://www.supinfo.com/cours/3AIT/chapitres/05-systeme-expert>
- [W4] http://www.brainkart.com/article/Characteristics-of-an-Expert-System_8929/
- [W5] https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_expert_systems.htm
- [W6] <https://www.memoireonline.com/09/10/3851/Realisation-dun-systeme-expert-pour-la-therapeutique-et-le-diagnostic-des-maladies-de-la-tuber.html>
- [W7] <https://hdd34.developpez.com/cours/artpoo/#L1.3.3>
- [W8] <https://web.maths.unsw.edu.au/~lafaye/CCM/poo/heritage.htm>
- [W9] [http://dictionnaire.sensagent.leparisien.fr/Polymorphisme%20\(informatique\)/fr-fr/](http://dictionnaire.sensagent.leparisien.fr/Polymorphisme%20(informatique)/fr-fr/)
- [W10] <https://fr.slideshare.net/AmineChkr/u-m-l-analyse-et-conception-objet>
- [W11] <https://theses.univ-oran1.dz/document/TH3168.pdf>
- [W12] <http://munier.perso.univ-pau.fr/temp/M4207C/TP1%20Intro%20Eclipse.pdf>

- [W13] https://fr.wikipedia.org/wiki/Java_Development_Kit
- [W14] <https://fr.wikipedia.org/wiki/StarUML>
- [W15] <https://www.clubic.com/telecharger-fiche220230-pacestar-uml-diagrammer.html>
- [W16] <https://www.emse.fr/~picard/cours/1A/java/livretJava.pdf>

Résumé

En raison de l'utilisation quotidienne de l'ordinateur, que ce soit dans le travail ou la vie personnelle, et en raison du grand nombre de travaux sur celui-ci, ce dernier est soumis à de nombreuses pannes, que ce soit au niveau du matériel ou des logiciels, et pour ces raisons, nous avons proposé de concevoir un système expert intelligent et efficace (**SEDIP**) qui nous aide à découvrir les types de dysfonctionnements de l'ordinateur, et ce système est considéré comme l'une des techniques de L'intelligence artificielle .

Mots clés : *L'ordinateur, Pannes, Matériel, Logiciel, Système expert intelligent, SEDIP, L'intelligence artificielle,...*

Abstract

Due to the daily use of the computer, whether in work or personal life, and due to the large number of works on it, the latter is subject to many failures, whether at the level hardware or software, and for these reasons, we have proposed to design an efficient intelligent system (SEDIP) that helps us find out the types of computer malfunctions, and this system is considered to be one of the techniques. Artificial intelligence.

Keywords: *Computer, Failures, Hardware, Software, Intelligent expert system, SEDIP, Artificial intelligence, ...*

ملخص:

بسبب الاستخدام اليومي للكمبيوتر سواء في العمل أو الحياة الشخصية ، وبسبب كثرة الأعمال عليه ، فإن هذا الأخير يتعرض للعديد من الإخفاقات ، سواء على المستوى الأجهزة أو البرامج ولهذا الأسباب اقترحنا تصميم نظام ذكي فعال (**SEDIP**) يساعدنا في معرفة أنواع الأعطال الحاسوبية ، ويعتبر هذا النظام من التقنيات. الذكاء الاصطناعي.

الكلمات المفتاحية: كمبيوتر ، اعطال ، اجهزة ، برامج، نظام ذكي خبير ، SEDIP ، الذكاء الاصطناعي

.....