



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option : Modèle Intelligence et Décision (M.I.D)

Thème

**Conception et Réalisation d'une application
pour la gestion des systèmes d'aide à la
décision médicale**

Réalisé par :

- **YAHLA MOHAMMED REDA**
- **MESSAOUD MOHAMMED**

Présenté le 1 Juin 2016 devant le jury composé de MM.

- *Mr HADJILA Fethallah* (Président)
- *Mr BENMOUNA Youcef* (Encadreur)
- *Mr BENAZZOUZ Mortada* (Examineur)
- *Mme CHAOUICHE RAMDAN Lamia* (Examineur)

Dédicace:

**A tous ceux qui, de près ou de loin, ont participé à notre éducation, nous
ont aidé dans les moments difficiles à sur**

Monter nos problèmes.

A ceux qui ont partagé avec nous nos bonheurs et nos

Soucis.

A notre très chères parents pour leurs sacrifices et leur

Soutien durant tout nôtres années d'étude.

A nos très chères soeurs.

Et toute la famille « YAHLA et MESSAOUD ».

A tout nos amis de plus proche au plus loin,

Hakim, Samir, Mustapha, gouthi,...et les autre chacun de

Sont nom.

A notre encadreur Mr Benmouna et

A notre promotion « MASTER INFORMATIQUE 2015/2016 »

YAHLA & MESSAOUD

REMERCIEMENTS

*Avant tout, nous remercions mon Dieu qui m'a donné la force, le
Courage et la volonté pour achever ce modeste travail, et je dis
« Al hamdou lillah ».*

*Qu'il nous est permis de remercier tout ceux qui nous ont
encouragés, soutenus et aidés à réaliser ce projet.*

*Nous remercions tout d'abord tous les professeurs de la faculté
de sciences des ingénieurs pour toutes les informations qu'il nous
a données durant notre période d'étude universitaire.*

*Nous adressons nos sincères mercis au chef de notre
département, ainsi que notre encadreur Mr Benmouna Youcef
pour toutes les informations et tous les conseils qu'il nous a
donné durant notre projet de fin d'étude.*

*Enfin, nous remercions tous les membres jury qui vont examiner
notre projet de fin d'études*

Table des matières

Table des matière	1
Table des Figures	3
INTRODUCTION GENERALE	5
CHAPITRE I : GENERALITES ET DEFINITIONS	8
1. Introduction	9
2. Système d'aide à la décision médicale	9
3. La typologie des systèmes d'aide à la décision médicale	9
3.1. Les systèmes d'aide indirecte à la prise de décision ou d'assistance documentaire	9
3.2. Les systèmes d'alerte ou de rappels automatiques	10
3.3. Les systèmes consultants	10
4. Les systèmes experts pour l'aide à la décision médicale	10
5. Les composants d'un système expert	11
5.1. Construction d'un système expert	12
5.1.1. Construction de la base de connaissance	12
5.2.2. Construction du moteur d'inférence	16
6. Quelques systèmes experts dans le domaine médical	19
7. Conclusion :	20
CHAPITRE II : CONCEPTION D'APPLICATION	21
1. Introduction	22
2. La modélisation du point de vue logique	22
2.1. UML (Unified Modelling language)	22
2.2. Diagramme cas d'utilisation	22
2.3. Diagramme de classes	25
2.4. Diagramme de séquence	27

3. Conclusion.....	33
CHAPITRE III : REALISATION DE L'APPLICATION	
MYEXPERT	34
1. Introduction	35
2. Outils utilisés.....	35
2.1. Outils de création du design	35
2.2. Outil de développement AGL :	36
2.3. Outil de l'apprentissage.....	37
3. Les composants GUI de MyExpert	41
4. Conclusion.....	46
CONCLUSION GENERALE	47
REFERENCES	49

Table des Figures

Figure 1 : Schéma d'un système expert.....	11
Figure 2 : Un ensemble flou de forme triangulaire	12
Figure 3 : Un système d'inférence floue	14
Figure 4 : Un exemple de défuzzification	15
Figure 5 : Moteur d'inférence à chaînage avant.....	17
Figure 6 : Algorithme du chaînage avant.....	17
Figure 7 : Moteur d'inférence a chaînage arrière	18
Figure 8 : Algorithme du chaînage arrière	19
Figure 9 : diagramme de cas d'utilisation.....	23
Figure 10 : Diagramme de classes.....	25
Figure 11 : Liste chaînées d'objets.....	26
Figure 12 : Diagramme de séquences (cas gérer la base faits part1).....	27
Figure 13 : Diagramme de séquences (cas gérer la base faits part 2).....	28
Figure 14 : Diagramme de séquences (cas gérer la base règles)	29
Figure 15 : Diagramme de séquences (cas gérer système expert).....	30
Figure 16 : Diagramme de séquences (cas faire des recherches et inférences).....	31
Figure 17 : Diagramme de séquences (cas charger une projet)	32
Figure 18 : Diagramme de séquences (cas faire l'apprentissage)	32
Figure 19 : Diagramme de séquences (cas sauvegarder un projet)	33
Figure 20 : Adobe PhotoShop CS6	35
Figure 21 : MyExpert design.....	36
Figure 22 : Eclipse IDE.....	36
Figure 23 : Fenêtre principale de FisPro	38
Figure 24 : Définition d'une entrée dans FisPro.....	39
Figure 25 : Définition d'une sortie dans FisPro.....	40
Figure 26 : Définition de règles dans FisPro.....	41
Figure 27 : Logo de l'application	41
Figure 28 : Menu Principal	42
Figure 29 : Gestion des fais.....	43
Figure 30 : Gestion des règles1.....	43
Figure 31 : Gestion des règles2.....	44

Figure 32 : Gestion de recherche1	44
Figure 33 : Gestion de recherche2	45
Figure 34 : A propos de MyExpert	45
Figure 35 : Fichier help.....	46

INTRODUCTION GENERALE

1-Introduction :

L'ampleur et la complexité croissante des connaissances médicales sont devenues de plus en plus nombreuses au fil des années avec les progrès de l'imagerie, le développement des tests biologiques, ainsi que la mise sur le marché de nouveaux médicaments et d'autres procédures thérapeutiques.

La mémoire humaine est limitée et le médecin ne peut avoir en tête l'ensemble des connaissances médicales dont il a besoin pour sa pratique quotidienne. De ce fait, une prise en charge des patients respectant les bonnes pratiques cliniques, nécessite que le médecin soit outillé pour réaliser ces tâches complexes.

2-Problématique :

Un système d'aide à la décision médicale (SADM) est un ensemble organisé d'informations, conçu pour assister le praticien dans son raisonnement en vue d'identifier un diagnostic et de choisir la thérapeutique adéquate, en opérant un dialogue entre l'homme et la machine.

La construction d'un système d'aide à la décision médicale (SADM) repose sur trois acteurs : le médecin, le cognitif et l'informatique.

Chacun a un rôle particulier à jouer dans les différentes étapes du projet, les tâches peuvent se répartir de la manière suivante : le cognitif recueille et analyse la connaissance fournie par le médecin, ensuite l'informatique choisit l'outil de développement et structure les données pour les entrer en machine.

On remarque que le médecin a besoin toujours des deux autres acteurs (le cognitif et l'informatique) pour construire un SADM.

3-Contribution :

Afin de satisfaire le besoin du médecin, Nous proposons dans ce travail un système flexible permettant une gestion efficace d'un système d'aide à la décision, sans l'intervention des deux acteurs (le cognitif et l'informatique), il permet de faciliter la tâche du médecin :

- Construire un SADM à partir d'une base d'apprentissage.
- Faire l'inférence sur ce système.
- Mise à jour des règles d'inférences.

- Sauvegarde d'un SADM.

4-Plan de lecture :

Après cette introduction, le premier chapitre est réservé pour quelques généralités et définitions sur les systèmes d'aide à la décision médicale et les travaux effectués dans ce domaine. Ensuite le deuxième chapitre est destiné à la conception et la modélisation. Le dernier chapitre montrera la réalisation de notre prototype et nous terminerons par des conclusions et perspectives.

CHAPITRE I : GENERALITES ET DEFINITIONS

1. Introduction

Les systèmes décisionnels devraient pouvoir assister de façon immédiate les médecins en leur proposant une synthèse dans leur pratique quotidienne, pourvu que celle-ci réponde réellement à leurs attentes. Plusieurs applications d'aide à la décision ont été développées dans ce domaine. Se sont destinées à soutenir le personnel de santé dans leurs prises de décisions avec l'utilisation de divers outils d'aide à la décision.

Dans le cadre de ce domaine nous nous sommes fixés l'implication des différentes techniques de fouille de données dans processus de prise de la décision.

2. Système d'aide à la décision médicale

Plusieurs définitions concernant les systèmes d'aide à la décision médicale ont été proposées. Un système d'aide à la décision médicale est un ensemble organisé d'informations, conçu pour assister le praticien dans son raisonnement en vue d'identifier un diagnostic et de choisir la thérapeutique adéquate, en opérant un dialogue entre l'homme et la machine. [1] Les systèmes d'aide à la décision médicale (SADM) sont « des applications informatiques dont le but est de fournir aux cliniciens en temps et lieux utiles les informations décrivant la situation clinique d'un patient ainsi que les connaissances appropriées à cette situation, correctement filtrées et présentées afin d'améliorer la qualité des soins et la santé des patients ».

3. La typologie des systèmes d'aide à la décision médicale

Il existe trois grandes catégories de systèmes d'aide à la décision médicale :

3.1. Les systèmes d'aide indirecte à la prise de décision ou d'assistance documentaire

L'objectif des systèmes d'assistance documentaire est de faciliter l'accès aux informations pertinentes en un temps record mais ces systèmes n'ont pas de méthode de raisonnement à proprement parler. [3]

L'accès aux résultats de laboratoire, la consultation des éléments importants du dossier médical du patient, les références bibliographiques (par exemple Medline) et les systèmes

de bases de données concernant les médicaments constituent des aides indirectes à la décision. Cette aide intervient pour faciliter l'appréciation d'une situation par le médecin. [4]

3.2. Les systèmes d'alerte ou de rappels automatiques

Certains systèmes permettent de rappeler au médecin des erreurs à ne pas commettre ou des éléments importants à prendre en compte pour la décision. Ils sont plus actifs et plus directement impliqués dans la décision médicale. L'assistance fournie n'est pas une aide au raisonnement ou à l'appréhension globale du cas du patient, mais plutôt un aide-mémoire fournissant une information utile et pertinente dans une situation facile à définir a priori [4].

Ces systèmes, comme les précédents, ne raisonnent pas véritablement. [3] Exemple, le rappel des valeurs normales des résultats d'examens de biologie et l'utilisation d'une typographie permettant d'attirer l'attention sur les valeurs anormales ou encore l'émission d'un message de mise en garde devant une association de médicaments déconseillée constituent une aide simple dont l'utilité est primordiale. [4]

3.3. Les systèmes consultants

Face à une situation médicale bien définie telle que : un diagnostic, une thérapie ou un pronostic, les systèmes consultants tentent d'émettre un avis de spécialiste [4]. Ces systèmes fournissent à l'utilisateur des conclusions argumentées selon les méthodes de raisonnement employées. La conception est intellectuellement plus satisfaisante que celles des systèmes n'utilisant pas de véritables processus de raisonnement. Donc les développeurs s'intéressent principalement à ce type de système où l'on note le plus de réalisations en matière de système d'aide à la décision. [3] Dans cette catégorie, on trouve les systèmes experts médicaux. [5]

4. Les systèmes experts pour l'aide à la décision médicale

Un système expert, selon la définition proposée par J.C. Pomerol, est un outil informatique d'intelligence artificielle, conçu pour simuler le savoir-faire d'un spécialiste, dans un domaine précis et bien délimité, grâce à l'exploitation d'un certain nombre de connaissances fournies explicitement par des experts du domaine.

Il permet de **modéliser** le raisonnement d'un expert, de manipuler des connaissances sous une forme déclarative, d'en faciliter l'acquisition, la modification et la mise à jour et de produire des explications sur la façon dont sont obtenus les résultats d'une expertise.

5. Les composants d'un système expert

Un système expert se compose d'une base de connaissances, d'un moteur d'inférences et de différentes interfaces qui lui permettent de communiquer avec son environnement.

[6]

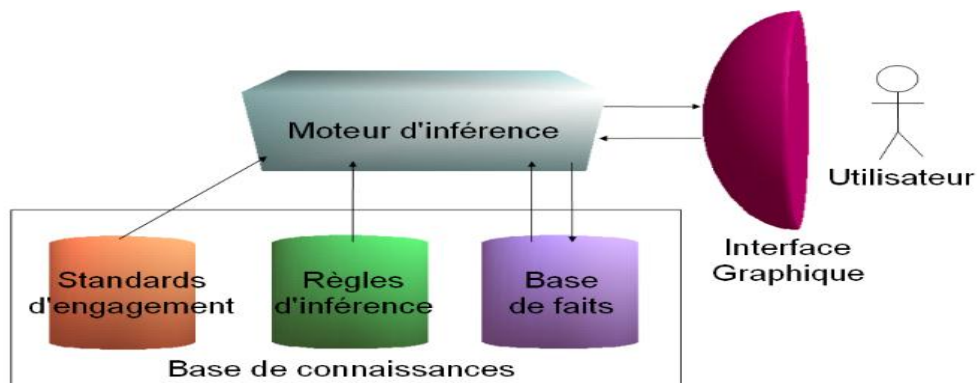


Figure 1 : Schéma d'un système expert

La base de connaissance

La base de connaissances se compose d'une base de faits et d'une base de règles.

Base de faits :

La base de faits est l'ensemble des propositions connues du système à un moment donné.

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.

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.

Le moteur d'inférence :

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. [2]

5.1. Construction d'un système expert

5.1.1. Construction de la base de connaissance

• **Ensemble flou** : Un ensemble flou est défini par sa fonction d'appartenance. Un point de l'univers, x , appartient à un ensemble, A avec un degré d'appartenance. $0 \leq \mu_A(x) \leq 1$.

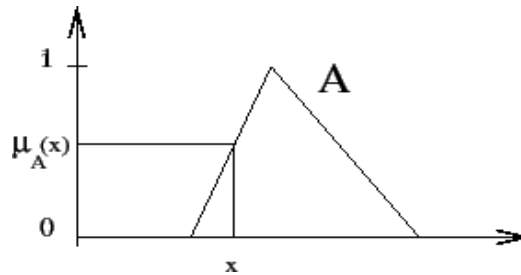


Figure 2 : Un ensemble flou de forme triangulaire

La figure 2 montre un ensemble de forme triangulaire.

• **Prototype d'un ensemble** : un point est un prototype d'un ensemble flou si son degré d'appartenance à cet ensemble vaut un. [7]

• **Opérateurs** :

- **ET** : opérateur de conjonction, noté \wedge , les plus employés sont le minimum et le produit.
- **OU** : opérateur de disjonction, les plus employés sont le maximum et la somme.
- **est** : la relation $x \text{ est } A$ est quantifiée par le degré d'appartenance de la valeur x au sous-ensemble flou A . [7]

• **Partitionnement** : Le découpage du domaine de définition d'une variable en sous-ensembles flous est appelé partitionnement. Ces ensembles sont notés A_1, A_2, \dots

• **Partition floue forte** : La partition de la variable X_i sera appelée une partition floue forte si $\forall x \in X_i, \sum_j \mu_{A_j^i}(x) = 1$

Exemple : un exemple ou individu est formé d'un vecteur d'entrée x de dimension P et, dans le cas général, d'un vecteur y , de dimension Q . [7]

• **Règle floue** : Une règle floue est de la forme **Si** je rencontre telle situation **Alors** j'en tire telle conclusion. La situation, appelée prémisse ou antécédent de la règle, est définie par une combinaison de relations de la forme $x \text{ est } A$ pour chacune des composantes du vecteur d'entrée. La partie conclusion de la règle est appelée conséquence, ou encore simplement conclusion. [7]

• **Les règles sont de deux types** :

◦ **Mamdani** : Une règle floue de type Mamdani, dont la conclusion est un ensemble flou, s'écrit :

$$SI \ x_1 \text{ est } A_1^i \ ET \ \dots \ ET \ x_p \text{ est } A_p^i \ ALORS \ y_1 \text{ est } C_1^i \ \dots \ ET \ y_q \text{ est } C_q^i$$

où A_j^i et C_j^i sont des ensembles flous qui définissent le partitionnement des espaces d'entrée et de sortie. [7]

◦ **Takagi-Sugeno** : Dans le modèle de Sugeno la conclusion de la règle est nette. Celle de

la règle i pour la sortie j est calculée comme une fonction linéaire des entrées : $y_j^i = b_{j0}^i + b_{j1}^i x_1 + b_{j2}^i x_2 + \dots + b_{jp}^i x_p$, également notée : $y_j^i = f_j^i(x)$.

• **Règle incomplète** : Une règle floue sera dite incomplète si sa prémisse est définie par un sous-ensemble des variables d'entrée seulement. La règle, $SI \ x_2 \text{ est } A_2^1 \ ALORS \ y \text{ est } C_2$ est incomplète car la variable x_1 n'intervient pas dans sa définition. Les règles formulées par les experts sont principalement des règles incomplètes. Formellement, une règle incomplète est définie par une combinaison implicite de connecteurs logiques ET et OU opérant sur l'ensemble des variables. Si l'univers de la variable x_1 est découpé en 3 sous-ensembles flous, la règle incomplète ci-dessus peut aussi s'écrire de la façon suivante :

SI (x_1 est A_1^1 OU x_1 est A_1^2 OU x_1 est A_1^3) ET x_2 est A_2^1 ALORS y est C_2
 [7]

- **Degré de vérité** : Pour une règle donnée i , son degré de vérité pour un exemple, également appelé poids, et noté w_i , résulte d'une opération de conjonction des éléments de la prémisse : $w_i = \mu_{A_1^i}(x_1) \wedge \dots \wedge \mu_{A_p^i}(x_p)$, où $\mu_{A_j^i}(x_j)$ est le degré d'appartenance de la valeur x_j à l'ensemble flou A_j^i .

- **Activité** : Un exemple active une règle, ou bien une règle active un exemple, si le degré de vérité de la règle pour l'exemple est non nul. [7]

- **Prototype d'une règle** : un exemple est un prototype d'une règle si son degré de vérité pour cette règle vaut un.

- **Système d'inférence floue (SIF)** : Un système d'inférence floue est formé de trois blocs comme indiqué sur la figure 3. Le premier, l'étage de fuzzification transforme les valeurs numériques en degrés d'appartenance aux différents ensembles flous de la partition. Le second bloc est le moteur d'inférence, constitué de l'ensemble des règles. Enfin, un étage de défuzzification permet, si nécessaire, d'inférer une valeur nette, utilisable en commande par exemple, à partir du résultat de l'agrégation des règles. Le nombre de règles du système est noté r .

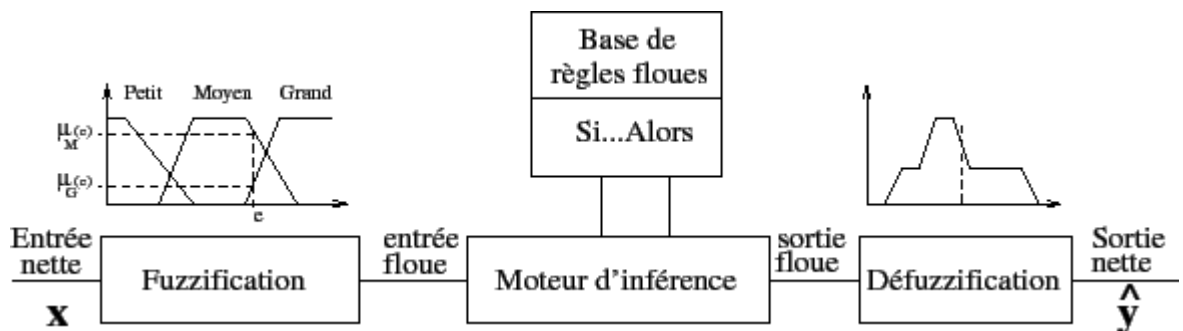


Figure 3 : Un système d'inférence floue

- **Sortie inférée par le système** : la valeur inférée, pour une entrée donnée, dépend bien entendu de la base de règles mais aussi des opérateurs d'agrégation et de défuzzification.

L'agrégation des règles est disjonctive, signifiant que chaque règle ouvre une nouvelle possibilité pour la sortie. Les deux principaux opérateurs sont le maximum et la somme. Les niveaux d'activation résultants sont, r étant le nombre de règles et m le nombre de termes linguistiques de la partition de la variable de sortie :

$$\forall j = 1, \dots, m$$

○ **max:**

$$W^j = \left\{ \max_r (w^r(x)) \mid C^r = j \right\}$$

$$\forall j = 1, \dots, m$$

○ **sum:**

$$W^j = \min \left(1, \left\{ \sum_r (w^r(x)) \mid C^r = j \right\} \right)$$

Plusieurs opérateurs de défuzzification sont disponibles. La figure 4 illustre la défuzzification pour un représentant des deux principales familles. [7]

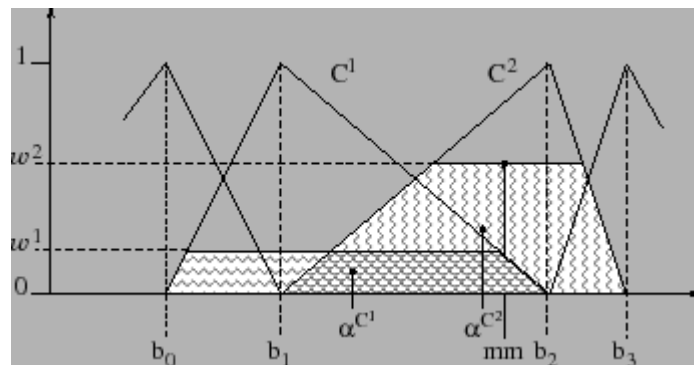


Figure 4 : Un exemple de défuzzification

Notons \hat{y}_i la valeur inférée pour l'exemple i .

Avec la moyenne des maxima, la sortie vaut $\hat{y}_i = mm$. Cet opérateur considère seulement le segment correspondant au niveau d'activation maximum. Aussi, il travaille principalement au sein d'un terme linguistique. D'autres valeurs que la moyenne est possible, comme le minimum ou le maximum des maxima. [7]

L'opérateur de pondération par les aires favorise l'interpolation entre termes linguistiques. où m est le nombre de sous-ensembles flous dans la partition, $\alpha = W^j$ est le niveau d'activation résultant de l'ensemble j , α^{C^j} est l'abscisse du centre de gravité de C_α^j , et C_α^j un nouvel ensemble flou. [7]

- **Apprentissage supervisé** : L'apprentissage supervisé consiste à induire des relations entre les entrées et la sortie, de dimension un, d'un système à partir d'un ensemble d'exemples.

5.2.2. Construction du moteur d'inférence

Il existe deux types d'algorithmes :

Chaînage avant

Le principe du chaînage avant représenté dans la figure 5 ci-dessous est simple, il requiert l'accès aux prémisses (standards d'engagement) afin de déclencher les règles d'inférence adéquates définies par les metarules. L'application des règles (évaluations) donnent des résultats, ceux-ci sont évalués (par les metarules) afin de savoir si l'on a accédé à une solution finale potentielle. Si c'est le cas, on arrête et cette solution est proposée. [8]

- Si c'est le cas, la solution est proposée à l'utilisateur. S'il la valide, la solution est enregistrée dans la base de faits comme *solution*, sinon comme simple résultat et on continue dans le cas suivant.

- Si cela n'est pas le cas ou si la solution est refusée, la solution est enregistrée dans la base de faits comme simple résultat et le moteur d'inférence tente d'y appliquer d'autres règles jusqu'à trouver une solution potentielle validée, ou jusqu'à ce qu'il n'y ait plus de règle. [8]

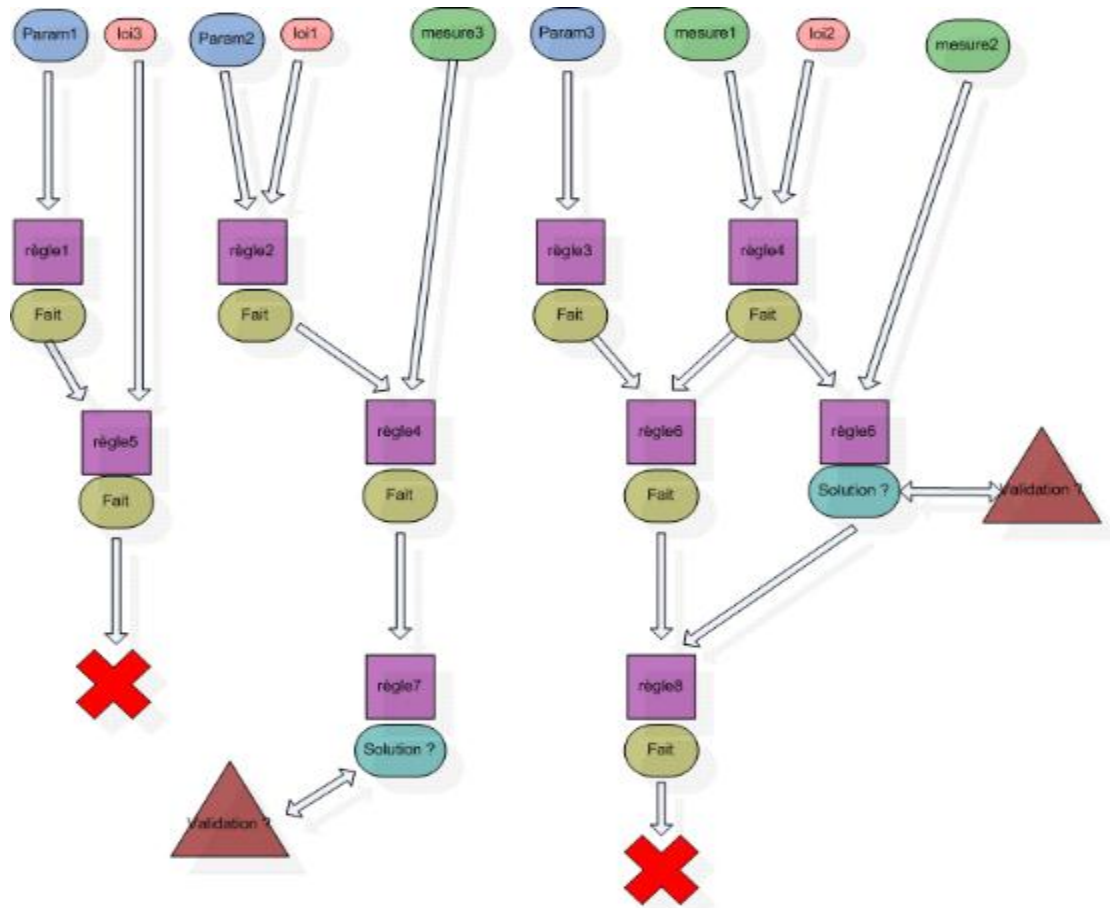


Figure 5 : Moteur d'inférence à chaînage avant

ENTREE : BF (BASE DE FAITS), BR (BASE DE REGLES), F (PROPOSITION A VERIFIER)
DEBUT
TANT QUE F N'EST PAS DANS BF ET QU'IL EXISTE DANS BR UNE REGLE APPLICABLE
FAIRE
PRENDRE LA PREMIERE REGLE APPLICABLE R
BR = BR - R (DESACTIVATION DE R)
BF = BF UNION CONCLUSION(R) (DECLENCHEMENT DE LA REGLE R, SA
CONCLUSION EST RAJOUTEE A LA BASE DE FAITS)
FIN TANT QUE
S'IL APPARTIEN A BF ALORS
F EST ETABLI (SUCCES)
SINON
F N'EST PAS ETABLI (ECHEC)
FIN.

Figure 6 : Algorithme du chaînage avant

Chaînage arrière

Le principe du chaînage arrière représenté dans la figure 7 ci-dessous est plus compliqué, il s'agit dans ce cas de partir d'un effet ou d'une solution et de tenter de remonter la chaîne afin de déterminer les causes d'un effet (fait). La procédure est à partir d'un fait, de déterminer, grâce aux metarules, les règles d'inférence qui auraient pu être à l'origine de ce fait et de déterminer les paramètres les plus probables. A partir de là, on analyse les paramètres :

- Si le paramètre est un fait enregistré dans la base de faits, c'est qu'il est le résultat d'une règle (évaluation). La procédure précédemment décrite est donc relancée.
- Si le paramètre n'est pas un fait de la base de faits, on en reste là.

On relève alors tous les faits et données retrouvés. Ils représentent les causes probables de la conséquence étudiée. [8]

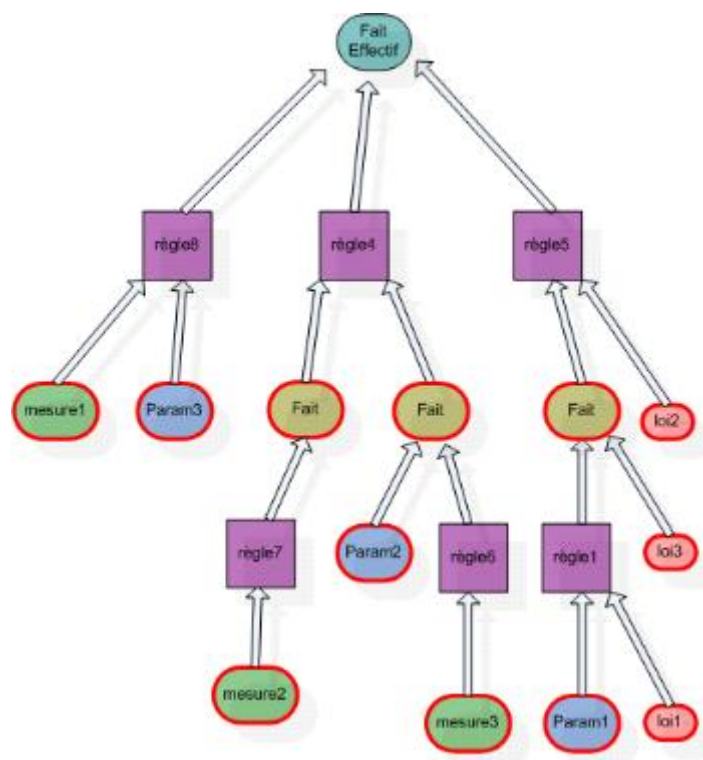


Figure 7 : Moteur d'inférence à chaînage arrière

```

ChainageArriere(BF, BR, F)
Si  $F \subset BF$  alors
    F est observable
Sinon
    F n'est pas observable
Fin Si

ChainageArriere(BF, BR, F)
DEBUT TRAITEMENT
    Soit R les règles R(i) de BR telles que  $F \subset Conclusion(R(i))$ 
    Si  $R = \{\emptyset\}$  alors
        Retourner( $\emptyset$ )
    Sinon
        Pour toutes les règles R(i) de R telles que  $F \subset Conclusion(R(i))$  Faire
            Si  $Condition(R(i)) \subset BF$  alors
                 $BR \leftarrow BR - \{R(i)\}$ 
                Pour toutes les règles R(j) telles que  $Conditions(R(j)) \subset BF$  Faire
                     $R \leftarrow R - \{R(j)\}$ 
                Fin Pour
            Retourner( $Conclusion(R(i))$ )
        Sinon
            Pour tout F de  $Conditions(R(i))$  Faire
                 $BF \leftarrow BF \cup ChainageArriere(BF, BR, F)$ 
            Fin Pour
        Fin Si
    Fin Pour
FIN TRAITEMENT

```

Figure 8 : Algorithme du chaînage arrière

6. Quelques systèmes experts dans le domaine médical

Les systèmes experts ont fait leur apparition dans certains domaines, particulièrement en médecine et ce dans plusieurs spécialités. Pourtant ces systèmes n'ont eu qu'un faible impact car ils étaient sous-utilisés par les praticiens qui leur reprochaient entre autres l'absence de méthodologie valide dans la constitution des bases de connaissances, citant quelque exemple :

MYCIN maladies infectieuses identification des microorganismes responsables des infections, conseil sur le choix d'un antibiotique

INTERNIST-I médecine interne diagnostic des problèmes complexes en médecine interne

PROTIS Diabétologie aide les médecins généralistes dans le traitement du diabète

SAM hypertensions artérielles diagnostic des hypertensions artérielles

SPHINX Endocrinologie aide au traitement du diabète NID, aide au diagnostique des icteres, proposition d'un modèle de simulation d'une consultation médicale

SES septicémies (infections généralisées) diagnostic et traitement des septicémies

MENINGE étude des méningites en pédiatrie étiologie et diagnostic des méningites bactériennes et virales

MEDICOTOXCONSILIUM toxicologie exogène diagnostic et traitement des intoxications exogènes

PATHFINDER chirurgie des ganglions lymphatiques diagnostic des maladies des ganglions lymphatiques

MYOSIS Physiologie diagnostic électromyographie

SETH intoxications médicamenteuses diagnostic et traitement des intoxications médicamenteuses

AES Bactériologie proposition d'un traitement antibiotique

Quelques inconvénients de système expert :

- Manque de flexibilité Efficacité limitée à un domaine spécifique
- Coûts de développement Difficulté de développement
- Mise à jour continuels requis Déshumanisation des opérations
- Dépendance accrue envers le système
- L'expert a besoin toujours un cognitifien.

7. Conclusion :

On remarque ces systèmes sont limités à un domaine spécifique, donc le médecin a besoin toujours d'un cognitifien et d'un informaticien pour construire un nouveau système, ce qui complique l'acquisition d'un autre système, à cause du coûts et le temps de développement, l'adaptation avec ce nouveau système.

Dans le but de faciliter et minimisé la complexité d'utilisation d'un système expert, nous proposons dans ce travail un système expert flexible et lisible à l'utilisation et rendre le médecin plus autonome envers les autres acteurs.

CHAPITRE II : CONCEPTION D'APPLICATION

1. Introduction

L'algorithmique qui était utilisé dans la programmation fonctionnelle ne pourrait pas suffire à lui seul. Le besoin d'avoir des méthodes ou langages pour la modélisation des langages orientés objet se faisait sentir. Ainsi plusieurs méthodes ou langages on vu le jour. En occurrence UML qui nous a permis de faire la conception de notre application.

2. La modélisation du point de vue logique

2.1. UML (Unified Modelling language)

Definition

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 est à la fois les notations et les concepts orientes 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 applique à toute science fondée sur la description d'un système. Dans l'immédiat, UML intéresse fortement les spécialistes de l'ingénierie système [9].

2.2. Diagramme cas d'utilisation

Description

Acteurs

- Utilisateur « Médecin ou Expert »
- My Expert « System Expert »
- FisPro «Outil d'apprentissage et System Expert »

Pré conditions

1. Installation du JRE (JAVA RUNTIME ENVIREMENT) dans la machine
2. Ajouter FisPro dans la liste des variables d'environnement du system d'exploitation

Buts et perspectives

1. Aide à la décision
2. Satisfaire l'utilisateur du coté lisibilité et efficacité

Ci-dessous est le diagramme de cas d'utilisation de l'application «My Expert »



Figure 9 : diagramme de cas d'utilisation

Cas défaillances et exceptions

1. Fichier invalide (soit .FAI ou .RGL ou .PRJ ou .FIS)
2. La liste de faits | ou règles est vide
3. Faits ou règles déjà existe (ex : Gestion de faits : ajouter un faits qui existe déjà)
4. Faits ou règles n'existe pas (ex : Gestion de faits : supprimer un faits qui n'existe pas)
5. FisPro est Introuvable (n'est pas dans la liste des variables d'environnement)

Etapes d'exécution

1. Gérer un system expert projet
 - 1.1. Gérer la base de faits (+V) [+V => plus Vérification system]
 - 1.1.1. Charger une base de faits
 - 1.1.2. Sauvegarder une base de faits
 - 1.1.3. Ajouter un fait
 - 1.1.4. Renommer un fait
 - 1.1.5. Supprimer un fait
 - 1.2. Gérer la base des règles (+V)
 - 1.2.1. Charger une base de règles
 - 1.2.2. Sauvegarder une base de règles
 - 1.2.3. Ajouter ou supprimer une règle
 - 1.2.4. Ajouter ou supprimer une liste de conséquences d'une règle
 - 1.2.5. Ajouter ou supprimer une liste de faits d'une règle
 - 1.2.6. Supprimer tous les règles
 - 1.3. Faire des recherches et inférences (+V)
 - 1.3.1. Sélection des faits initiaux
 - 1.3.2. Sélection du type de recherche (avant ou arrière)
 - 1.3.3. Le system infère les résultats
 - 1.3.4. Affichage des résultats dans le menu des résultats
2. Charger un projet (+V)
 - 2.1. Gérer un system expert
3. Charger un fichier .FIS (+V)
 - 3.1. Gérer un system expert
4. Lancer FisPro pour faire l'apprentissage (+V)
5. Consulter le fichier Help

2.3. Diagramme de classes

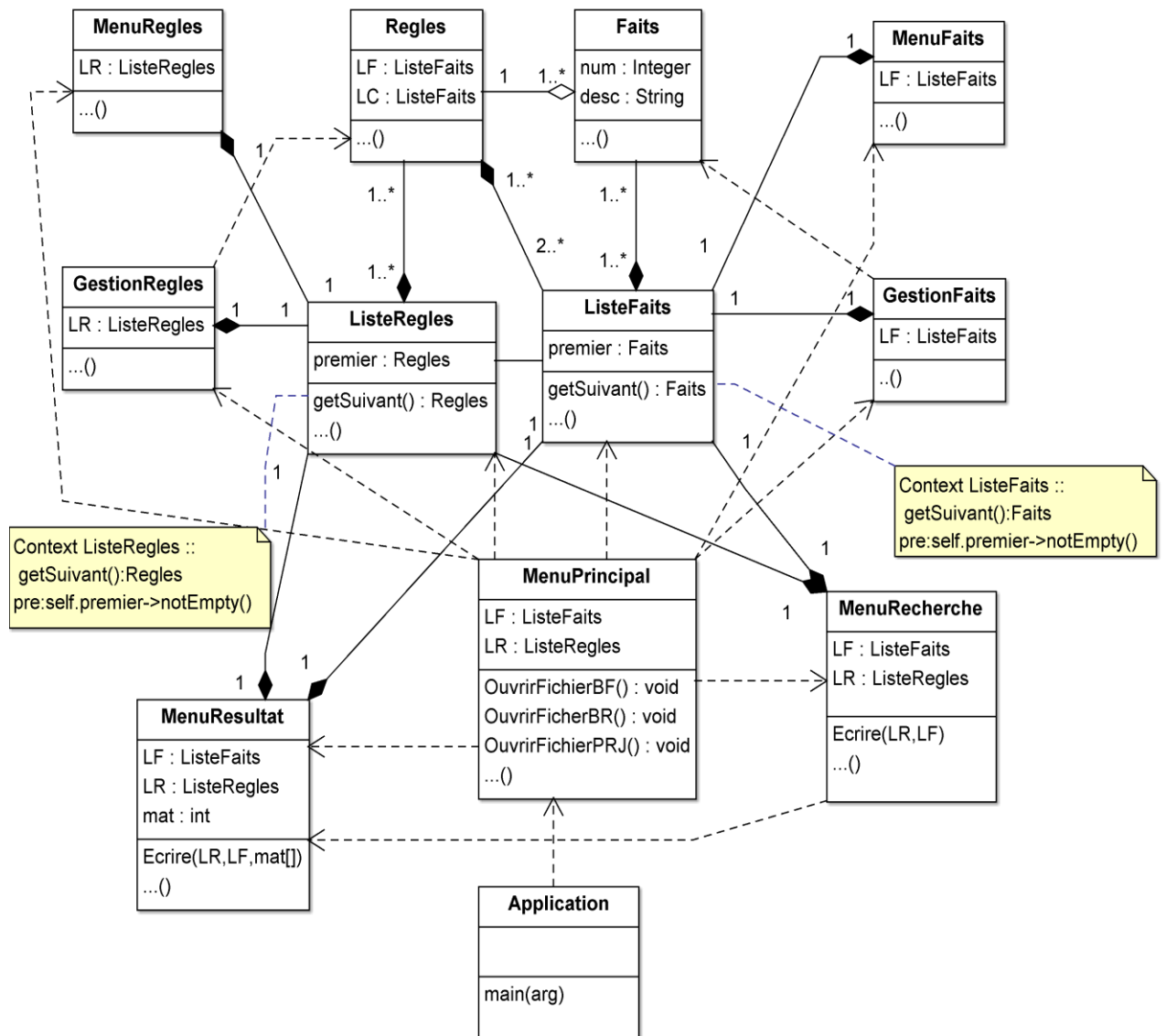


Figure 10 : Diagramme de classes

Description

Ce diagramme représente les classes nécessaires pour assurer un bon fonctionnement du système à mettre en œuvre, le système est composé de plusieurs menu, chaque menu est contenu deux composant essentiel sont la liste des faits, et la List des règles, c'est deux dernier est nul le système est en état élu.

Architecture utilisée

L'idée est de modéliser la base de faits et la base de règles en utilisant une structure algorithmique adaptée qui facilitera la mise en place du moteur d'inférence.

Solution : Représentation sous forme liste chaînées d'objets

Il y a plusieurs façons de représenter les listes chaînées d'objets. L'idée de base est d'utiliser un enchaînement des objets :

- chaque objet (fait ou règle) contient un élément suivant (fait ou règle) dans la liste (Liste Faits ou Liste Règles).
- chaque objet contient une référence vers l'objet suivant, sauf le dernier objet de la liste (qui contient une référence nulle).
- la liste donne l'accès au premier objet (fait ou règle), le reste de la liste est accessible en passant d'objet en objet, suivant leur enchaînement.

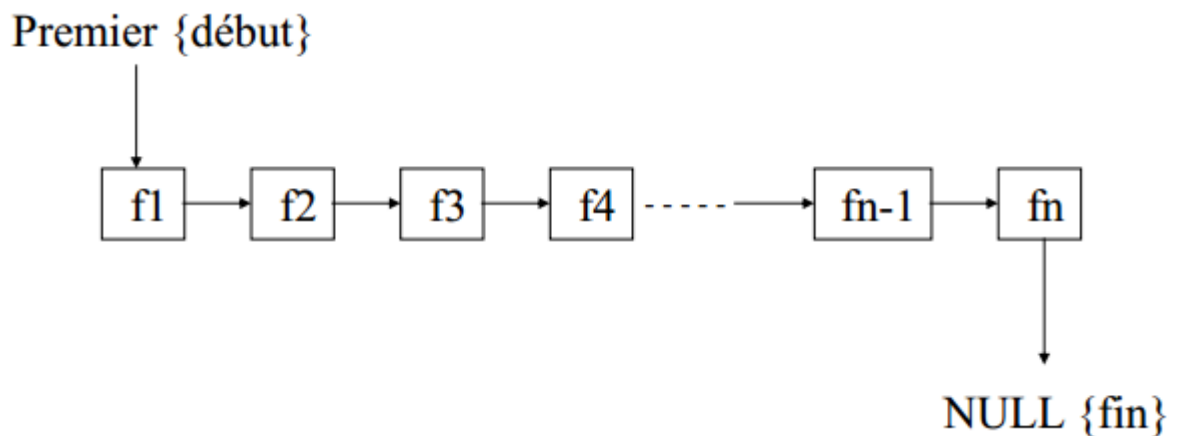


Figure 11 : Liste chaînées d'objets

2.4. Diagramme de séquence

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

1. Cas gérer la base des faits

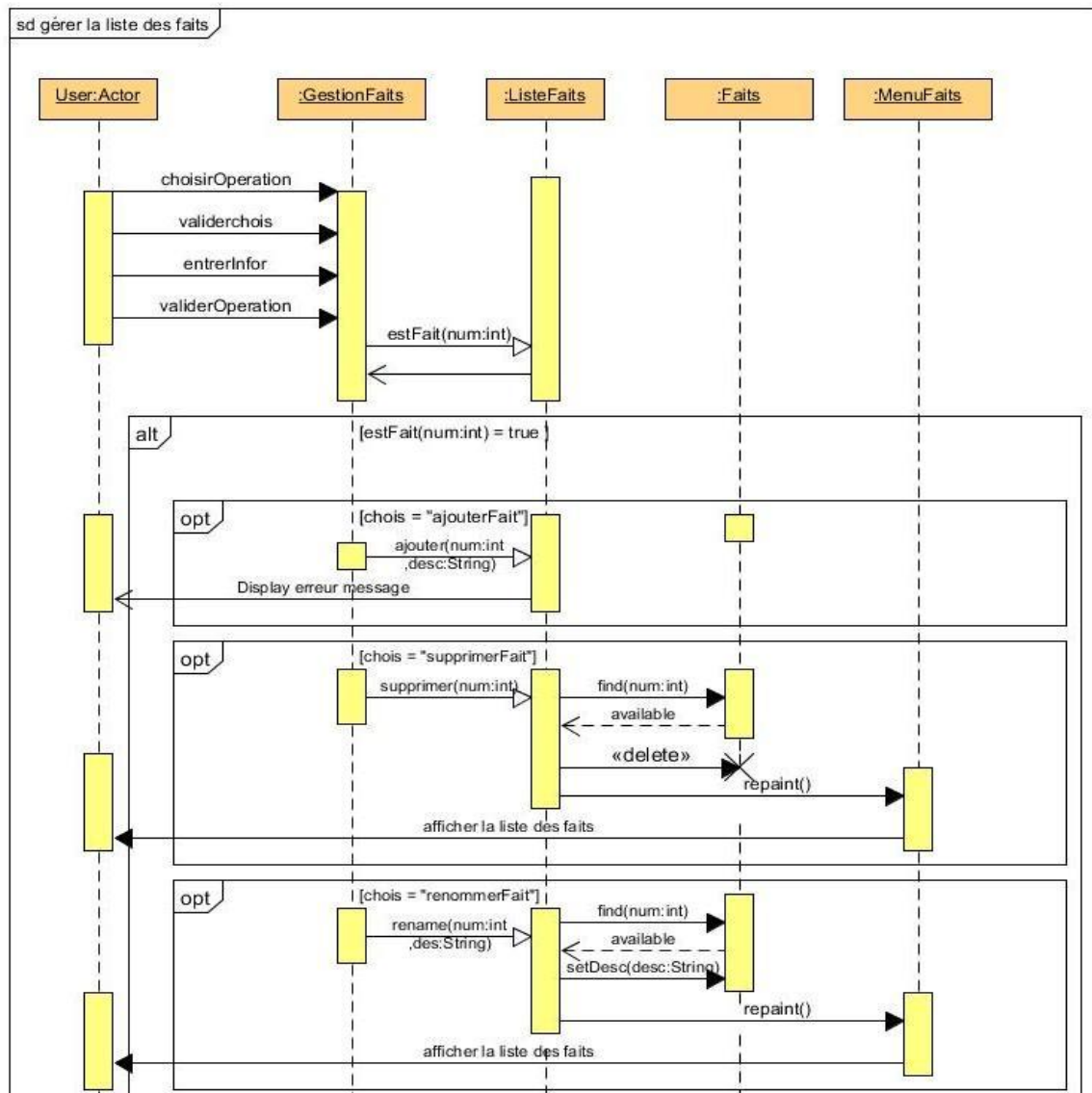


Figure 12 : Diagramme de séquences (cas gérer la base faits part1)

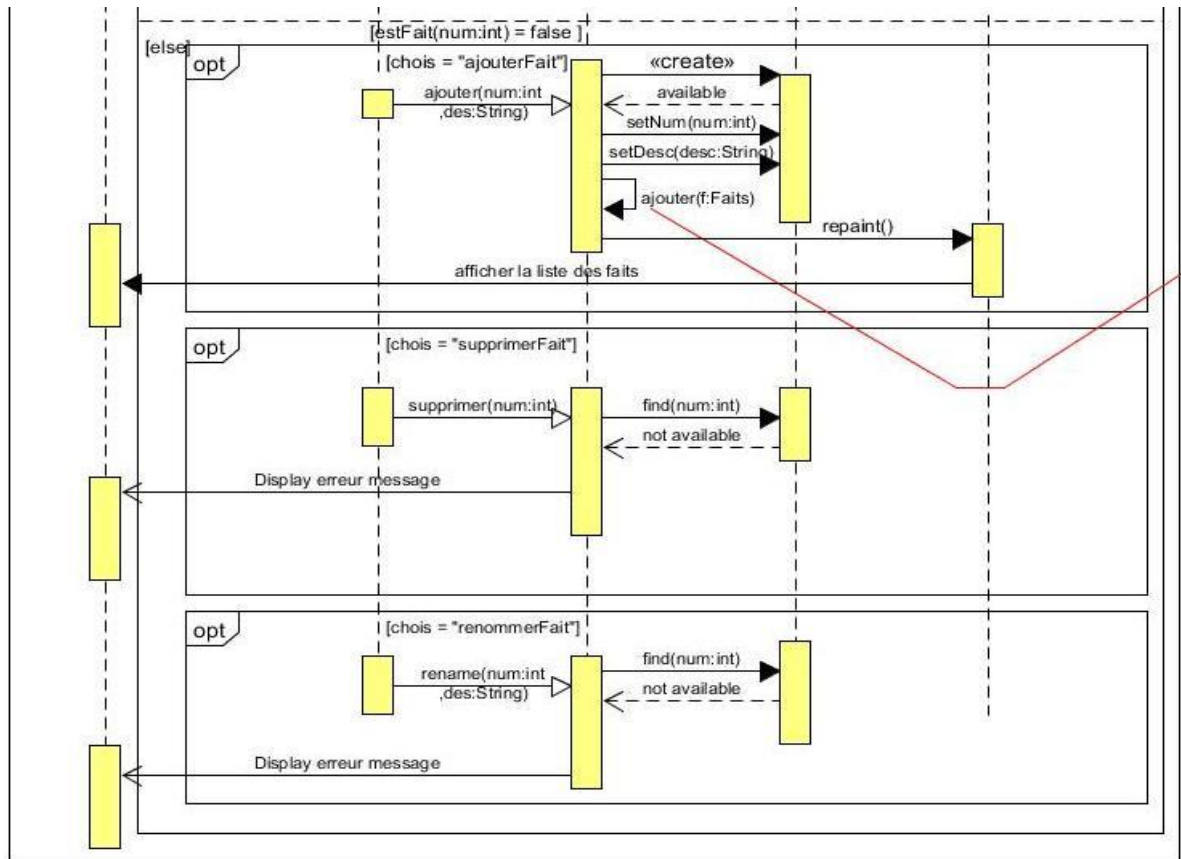


Figure 13 : Diagramme de séquences (cas gérer la base faits part 2)

Le scénario

Dans le processus de gestions des faits on a l'acteur responsable qui interagit avec le système.

- Pour chaque opération (ajouter, supprimer, renommer) l'utilisateur doit saisir les information essentiel avant de valider l'opération sélectionné.
- Dès que l'utilisateur valide l'opération, Le système va passé par un test si le faits spécifier par le numéro entré déjà existe, et il va générer des exception en cas de défaillance par exemple :

Ajouter un fait qui déjà existe ou supprimer un fait qui n'existe pas.

Cas gérer la base des règles

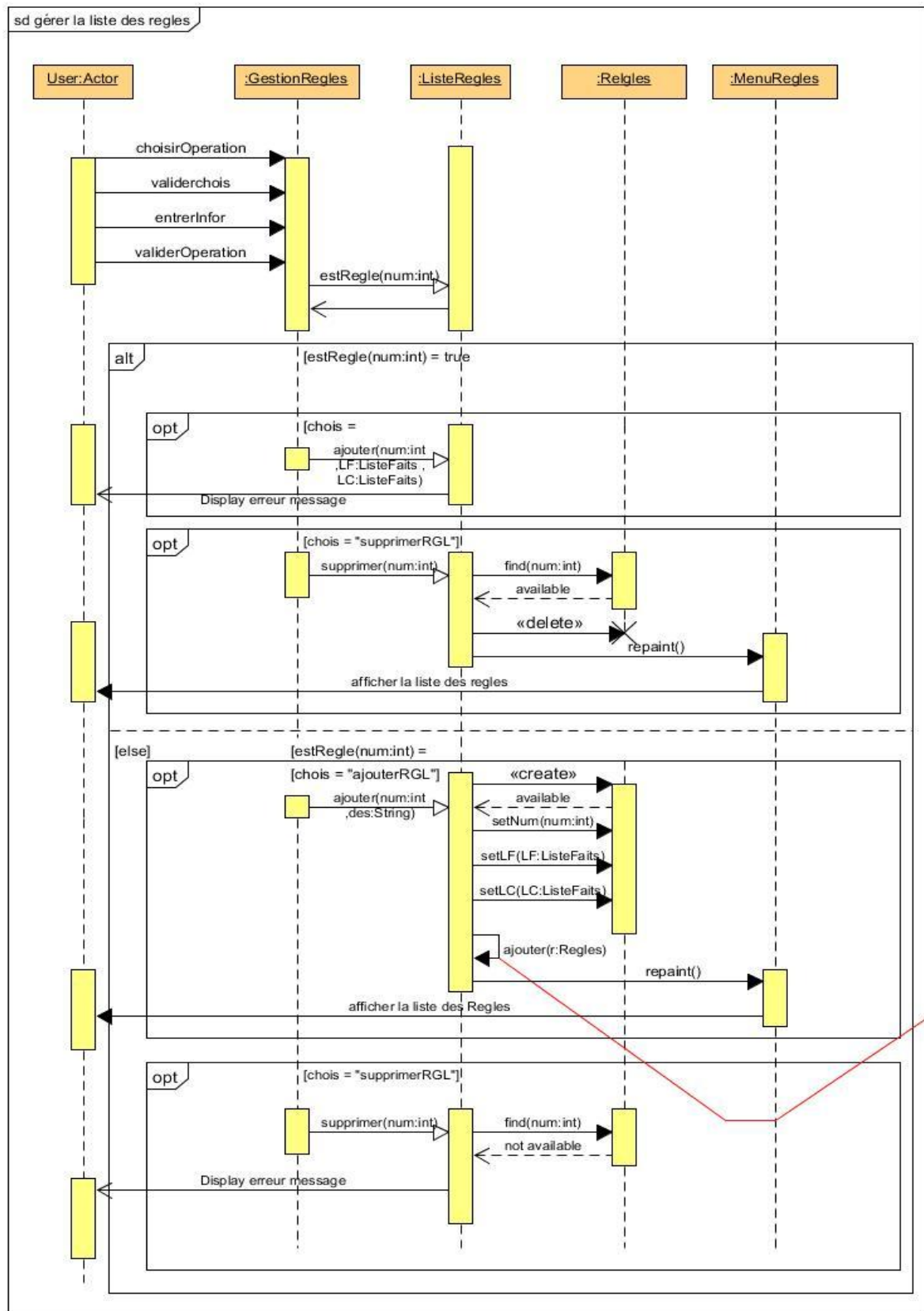


Figure 14 : Diagramme de séquences (cas gérer la base règles)

Le scénario

De même avec le processus de gestion des règles on a l'acteur responsable qui interagit avec le système.

- Pour chaque opération (ajouter, supprimer) l'utilisateur doit saisir les informations essentielles avant de valider l'opération sélectionnée.
- Dès que l'utilisateur valide l'opération, le système va passer par un test si la règle spécifiée 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.

Cas gérer un système expert

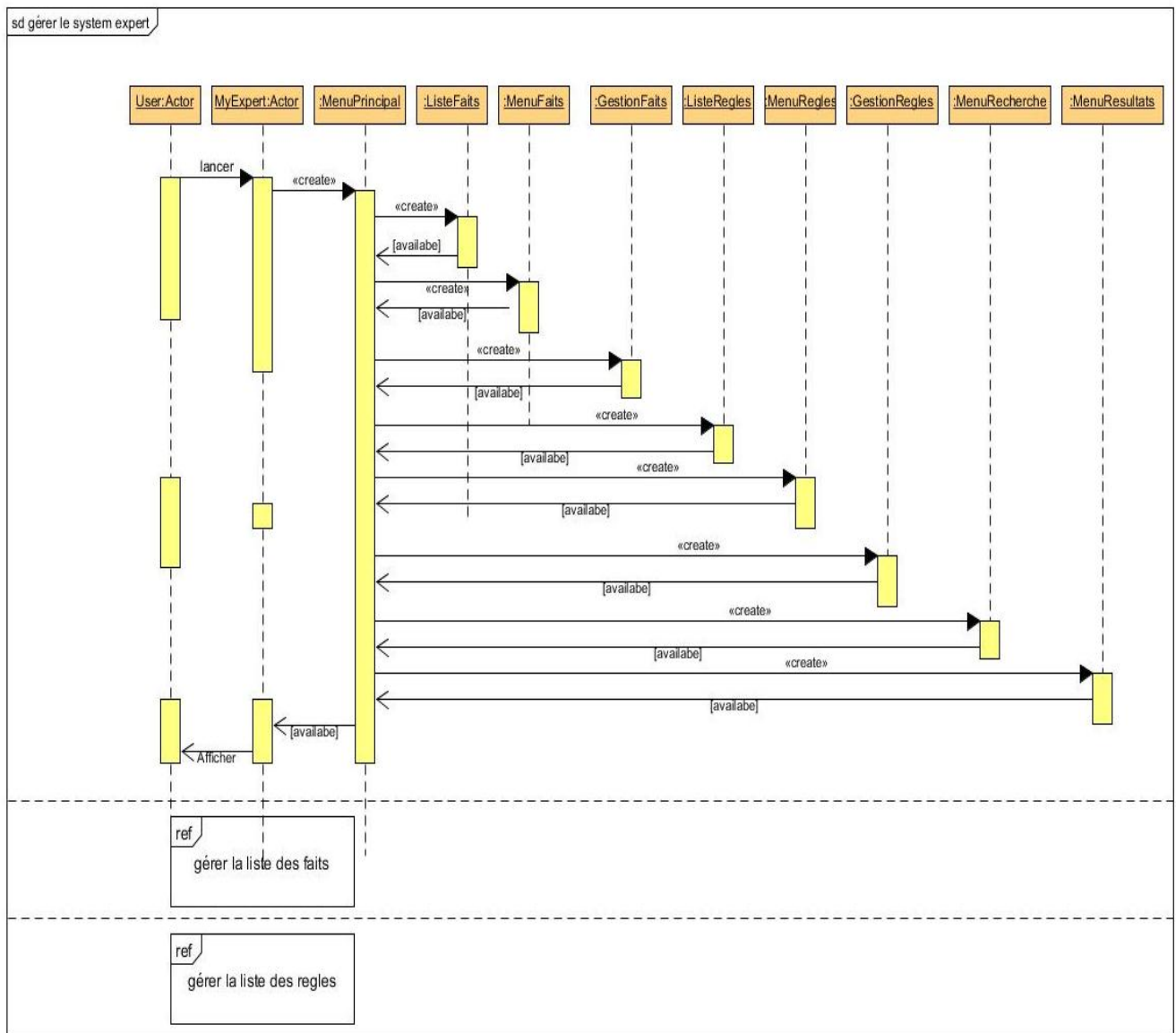


Figure 15 : Diagramme de séquences (cas gérer système expert)

Le scénario

Après l'acteur lance l'application, un nouveau projet créé automatiquement en début. Pour créer un nouveau projet il faut d'abord initialiser la liste de faits et la liste de règle, et après initialiser tous les composants qui contiennent ces deux listes.

Ensuite l'utilisateur doit gérer la liste des faits et des règles (voir figure GF et figure GR).

Cas faire des recherches et inférences

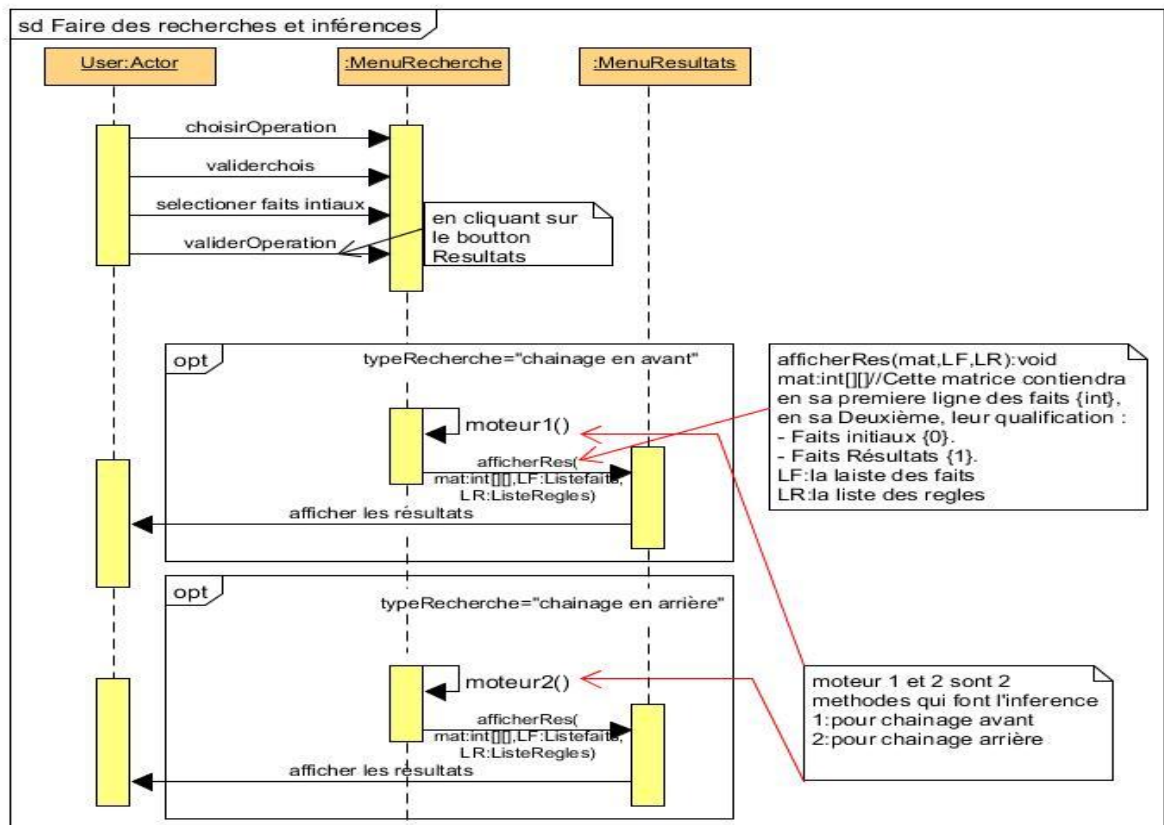


Figure 16 : Diagramme de séquences (cas faire des recherches et inférences)

Le scénario

Pour que L'acteur peut lancer une recherche, 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.

Cas charger un projet:

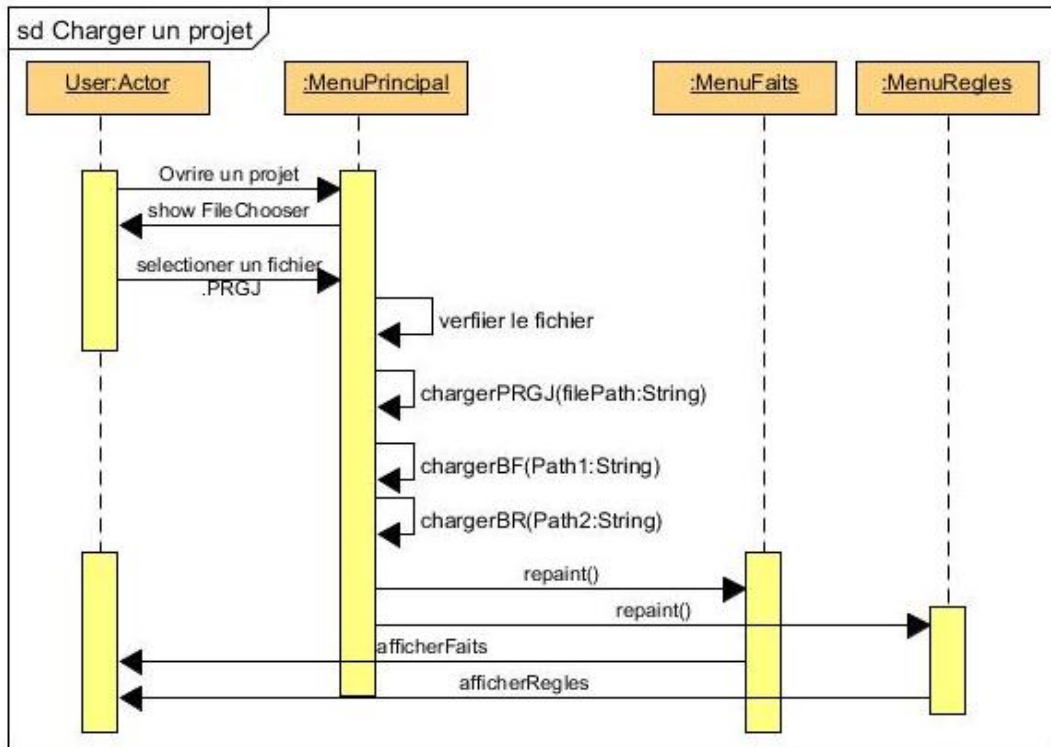


Figure 17 : Diagramme de séquences (cas charger une projet)

Le scénario :

Le chargement d'un projet doit être passer par le chargement de son base de connaissances correspondante.

Cas faire l'apprentissage

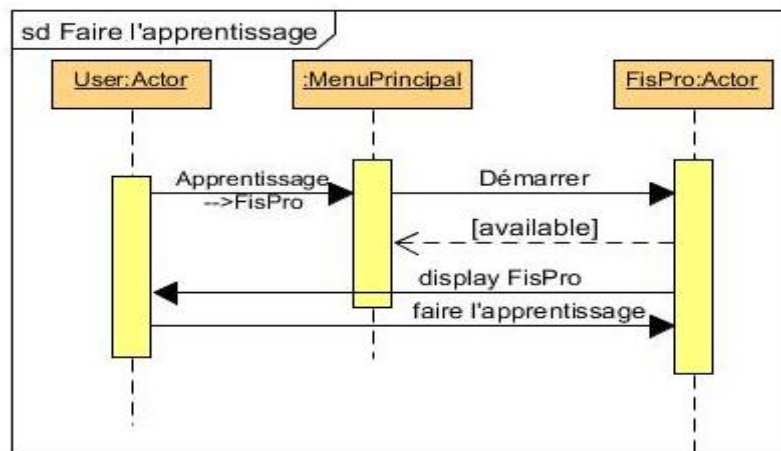


Figure 18 : Diagramme de séquences (cas faire l'apprentissage)

Le scénario

L'utilisateur peut utiliser l'outil FisProp pour faire la construction de la base de connaissance, l'outil peut être exécuté depuis le menu principal de l'application.

Cas sauvegarder un projet:

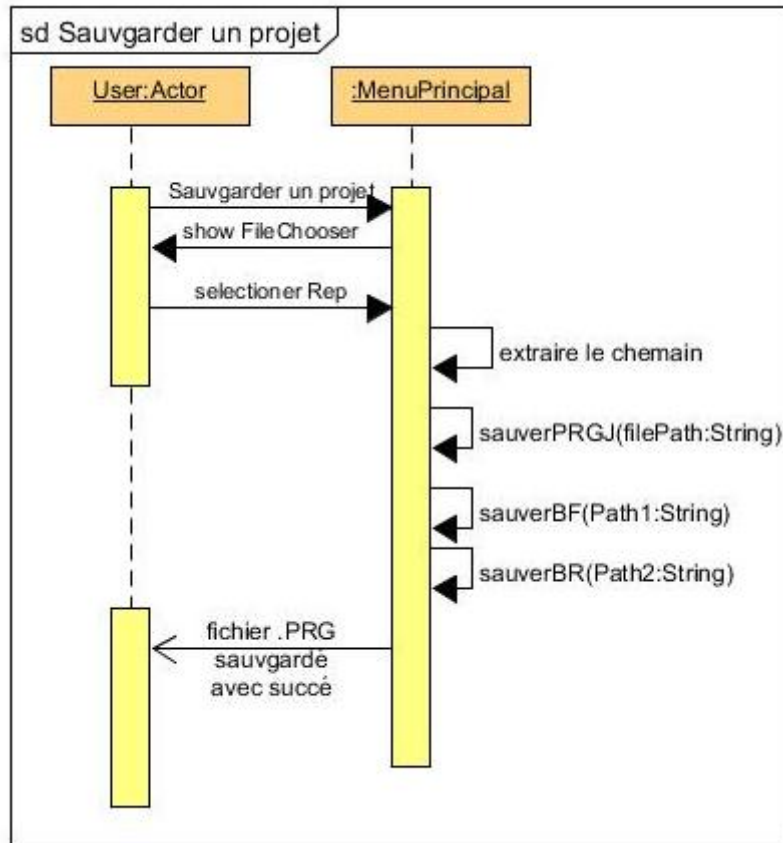


Figure 19 : Diagramme de séquences (cas sauvegarder un projet)

Le scénario

Pour stocker ou enregistrer un projet, il faut enregistrer d'abord son base de connaissance correspondante (base faits, base règles).

3. Conclusion

Dans ce chapitre on a modélisé le point de vue logique qui va répondre à la problématique mentionnée dans le chapitre précédent, et à l'aide des diagrammes du langage UML, on a achevé une bonne conception de l'application.

CHAPITRE III : REALISATION DE L'APPLICATION « MYEXPERT »

1.Introduction

Dans ce chapitre on va essayer de donner un bref aperçu sur quelques outils utilisés dans la réalisation de notre application, puis présenter les résultats de notre travail et finir par une petite conclusion.

2. Outils utilisés

2.1. Outils de création du design

Tous les ressources graphiques (images, icônes..) utilisés dans notre application sont créées au niveau du l’outil puissant Adobe Photoshop CS6.



Figure 20 : Adobe PhotoShop CS6

Photoshop est un logiciel de traitement d'image professionnel qui permet de faire de la retouche photo, du dessin (digital painting), du graphisme ou encore du web-design. Il est utilisé dans de nombreux domaines : les agences de publicité, les photographes, les graphistes, le cinéma ou encore les particuliers.[10]

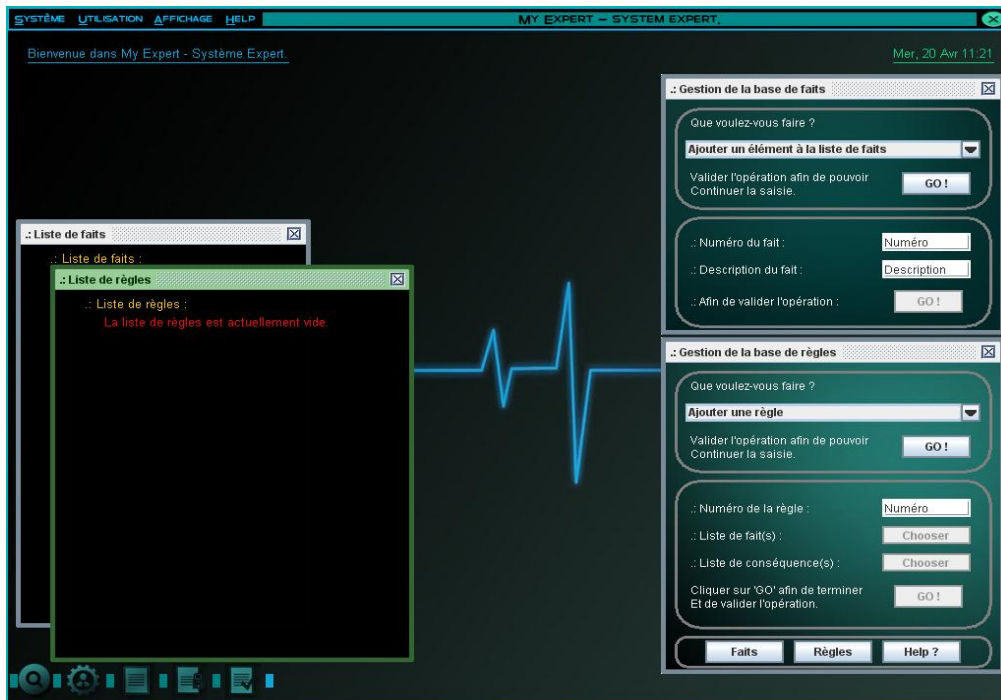


Figure 21 : MyExpert design

2.2. Outil de développement AGL :

Plateforme : notre application est réalisée sous la plateforme java.

Environnement de développement : « Eclipse Luna »

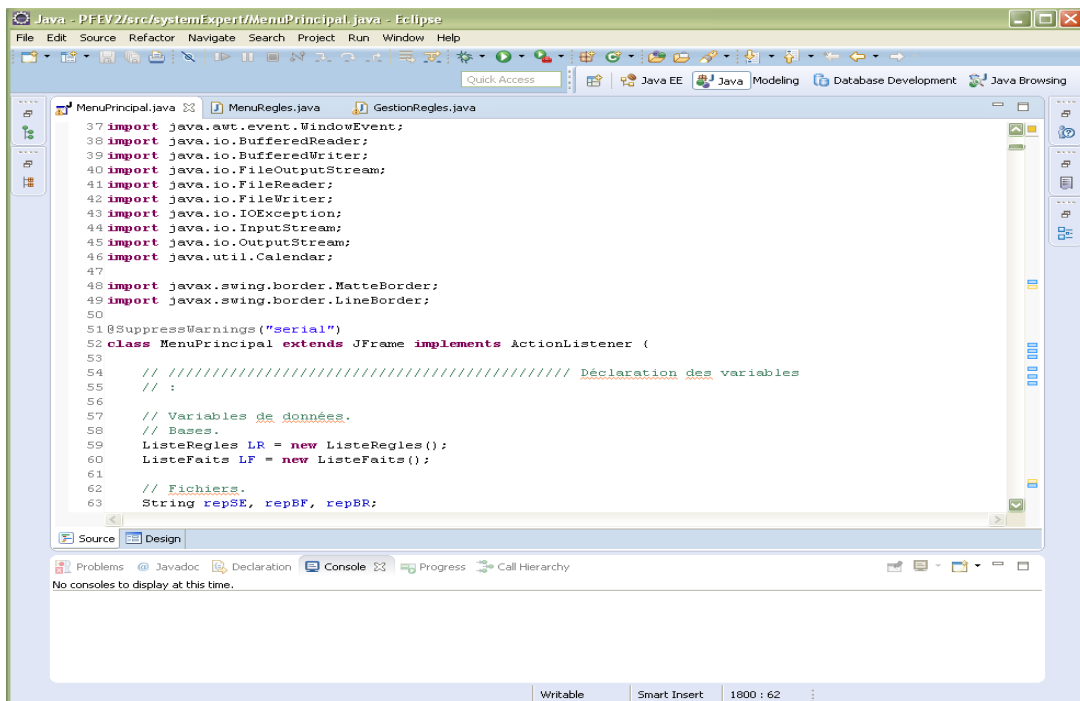


Figure 22 : Eclipse IDE

Eclipse est un projet, décliné et organisé en un ensemble de sous projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

2.3. Outil de l'apprentissage

2.3.1 Description de l'outil FisPro

FisPro (Fuzzy Inference System Professional) permet de créer des systèmes d'inférence floue, en particulier il est utilisé pour la simulation d'un système physique ou biologique. Il fonctionne à partir des règles de raisonnement floues, qui ont l'avantage de gérer la progressivité des phénomènes.

L'implémentation faite dans FisPro permet tout d'abord de créer directement des systèmes à partir de la connaissance experte d'un domaine.

2.3.1 Créer un système simple

L'exemple choisi crée un système très simple : 1 entrée, 1 sortie et 3 règles.

L'entrée est le degré du parfum, la sortie est son prix. Les règles font évoluer le prix en fonction du degré.

- Commencer par choisir l'option *Nouveau* du menu Sif
- Le nom par défaut Nouveau SIF s'affiche dans le champ *Nom*. Il est modifiable par simple saisie. Donnes-lui le nom coop.
- La conjonction est l'opérateur de combinaison des prémisses de la règle. C'est par défaut le produit.

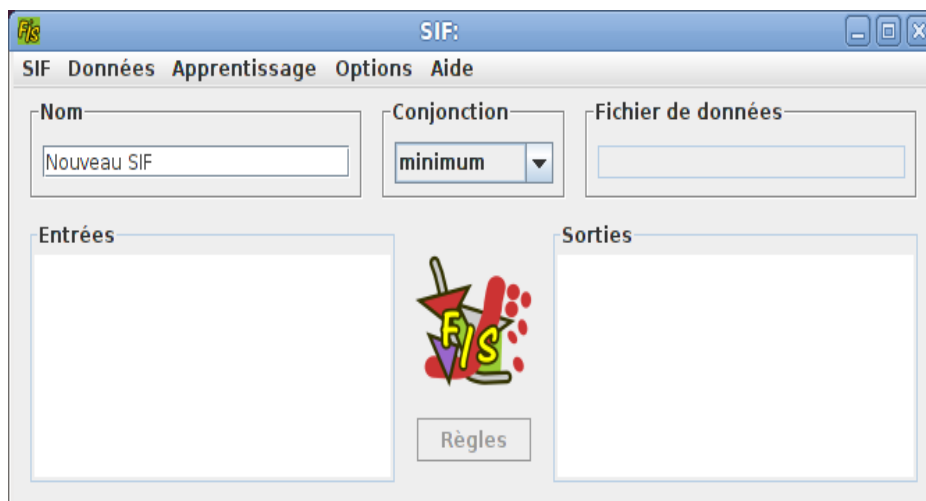


Figure 23 : Fenêtre principale de FisPro

2.3.3 Définir une entrée

Pour ajouter une entrée, on utilise soit l'option Nouvelle Entrée du menu SIF, soit un clic droit sur la zone Entrées de la fenêtre principale. La fenêtre Entrée apparaît.

Une entrée est caractérisée par son domaine, et sa partition floue, c'est-à-dire les sous-ensembles flous (SEF) qui la composent.

Elle peut être active (par défaut) ou inactive.

Donnez-lui le nom Degré.

- Le domaine de variation d'une entrée est par défaut [0,1].

Pour le changer, choisir le menu Domaine de la fenêtre Entrée, et entrer les nouvelles valeurs du domaine de variation : 9 et 14.

- La méthode la plus rapide, dans ce cas, pour définir la partition est l'option Grille irrégulière du menu SEF, avec le nombre de SEF correspondant à la finesse voulue pour les labels linguistiques (3 par défaut). Cette option permet de donner la position des sommets de chaque SEF : choisir 11,5, 12 et 12,5.

- Les SEF s'affichent dans la partie inférieure de la fenêtre : demi trapèzes aux extrémités du domaine, et triangles ailleurs.

- Pour la clarté du SIF, les noms des SEF sont importants, car ils apparaissent dans les règles.

- SEF 1 : nom *Faible*, sommets 9, 11,5 et 12

- SEF 2 : nom *Moyen*, sommets 11,5, 12 et 12,5
- SEF 3 : nom *Elevé*, sommets 12, 12,5 et 14

On obtient la partition ci-dessous.

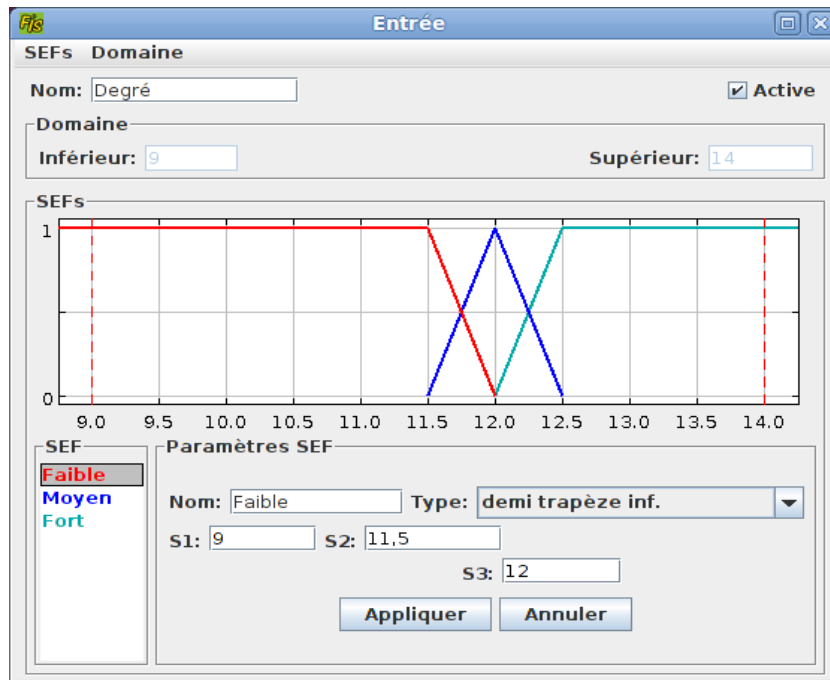


Figure 24 : Définition d'une entrée dans FisPro

Définir une sortie

Pour ajouter une sortie, on utilise soit l'option Nouvelle Sortie du menu SIF, soit un clic droit sur la zone Sortie de la fenêtre principale. La fenêtre Sortie apparaît.

On saisie le nom Prix. Une sortie est caractérisée avant tout par son domaine, et sa nature : sortie nette ou floue.

La nature est liée au raisonnement mis en œuvre dans le SIF.

- **Nature Nette** : Avec une sortie nette, la conclusion des règles peut être une valeur numérique quelconque.
- **Nature Floue** : Avec une sortie floue, la conclusion des règles est le label linguistique d'un SEF, par exemple *Petit, Moyen ou Grand*.

- **Valeur par défaut** : c'est la valeur que prendra le résultat de l'inférence, pour cette sortie, au cas où aucune règle n'est activée.
- **Défuzzification et disjonction** : choix liés à la façon d'agréger les conclusions des règles.
- **classif** : cocher cette case pour arrondir le résultat de l'inférence à la classe (valeur discrète) la plus proche.

Les classes possibles sont limitées aux valeurs des conclusions des règles.

Figure 25 : Définition d'une sortie dans FisPro

Définir une règle

Pour créer les règles, on doit cliquer sur la case Règles de la fenêtre principale. La fenêtre Règles s'affiche.

Dans cette fenêtre, pour ajouter une règle, on utilise soit l'option Nouvelle Règle du menu SIF, soit un clic droit sur la colonne Règle. Cliquer successivement sur les colonnes représentant les variables, pour choisir les labels qui interviendront dans la règle, ou pour entrer la valeur numérique (sortie nette).

Règle	Active	SI Degré	ALORS Prix
1	<input checked="" type="checkbox"/>	Faible	200
2	<input checked="" type="checkbox"/>	Moyen	300
3	<input checked="" type="checkbox"/>	Fort	600

Figure 26 : Définition de règles dans FisPro

3. Les composants GUI de MyExpert

1.1. Logo : Y&M réfère aux noms des développeurs (Yahla & Messaoud)



Figure 27 : Logo de l'application

Menu principal :



Figure 28 : Menu Principal

Le menu principal englobe tous les menus, on peut juste les montrer depuis les boutons de la barre d'outils au dessous, et depuis ce menu l'utilisateur peut faire la gestion d'un système expert, et à l'aide de la barre des menus au dessus il peut faire le stockage de données pour chaque projet ou charger une base ou un projet.

1.2. Menus de gestion de la base des faits :

Pour la gestion de la base des faits on a spécifié deux menus :

- Un menu pour l'affichage de la base.
- Un menu pour faire la gestion.

la [Figure 29] ci-dessous visualise deux menus, le premier dans les cas où la liste de faits est vide, et dans le deuxième la liste est remplie.

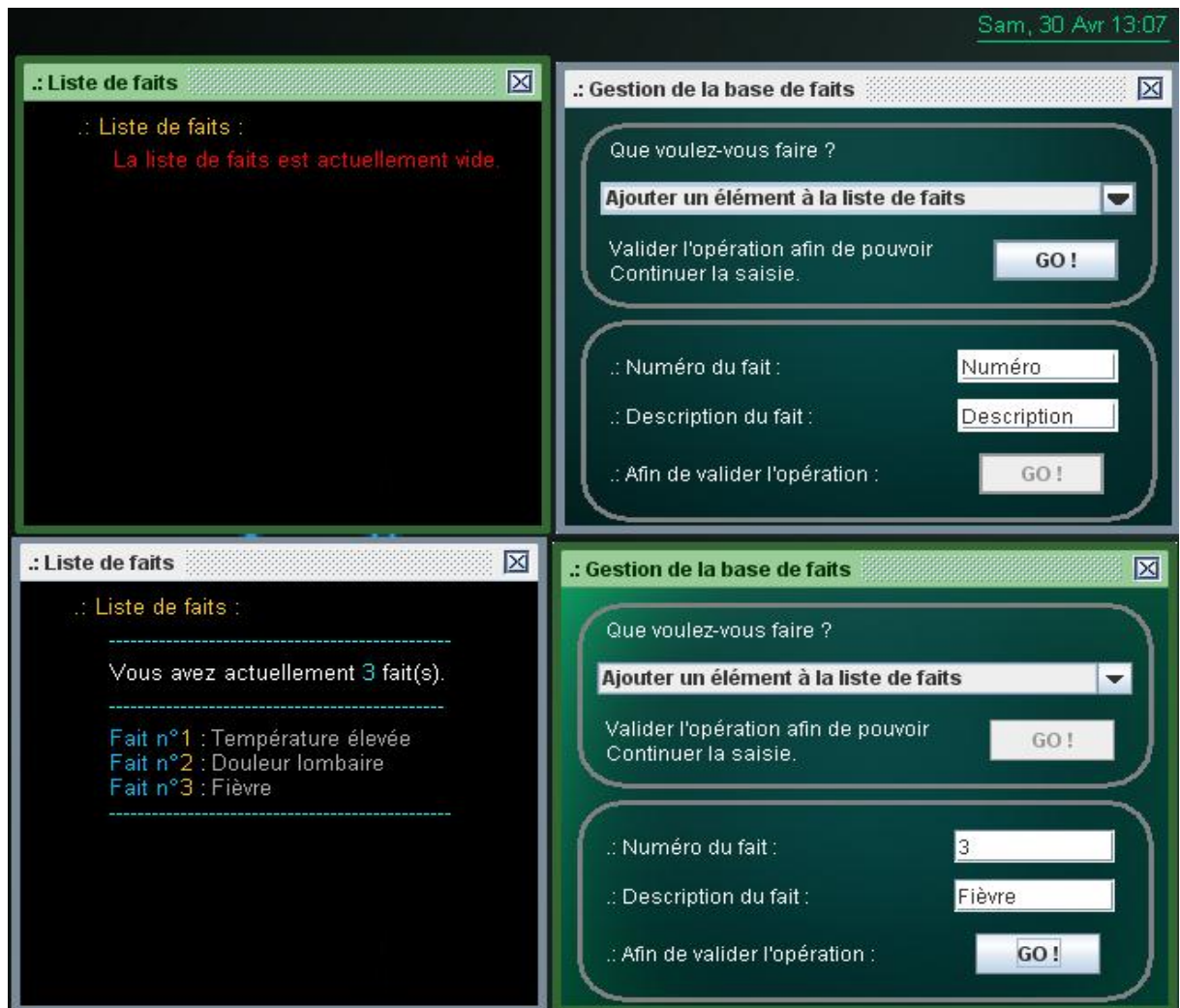


Figure 29 : Gestion des faits

1.3. Gestion de la base des règles :

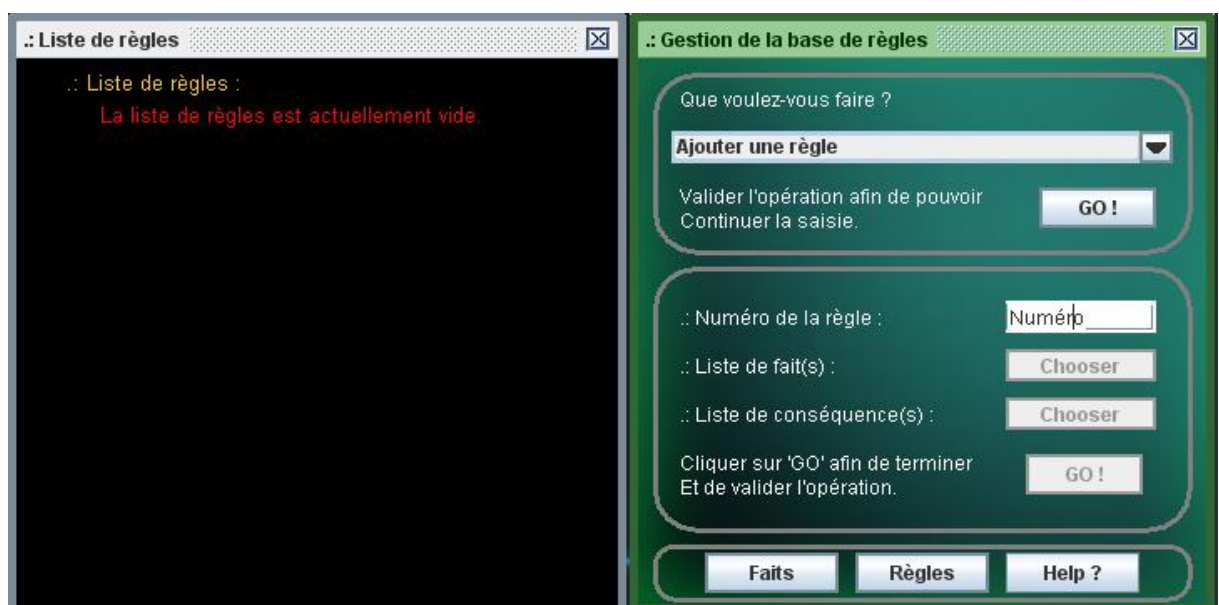


Figure 30 : Gestion des règles1

Pour la gestion de la base des règles il y a deux menus:
 Un menu pour l'affichage de la base des règles et un autre pour la gestion.
 la [Figure 30] visualise une liste de règles est vide.
 La [Figure 31] visualise une liste contient des règles

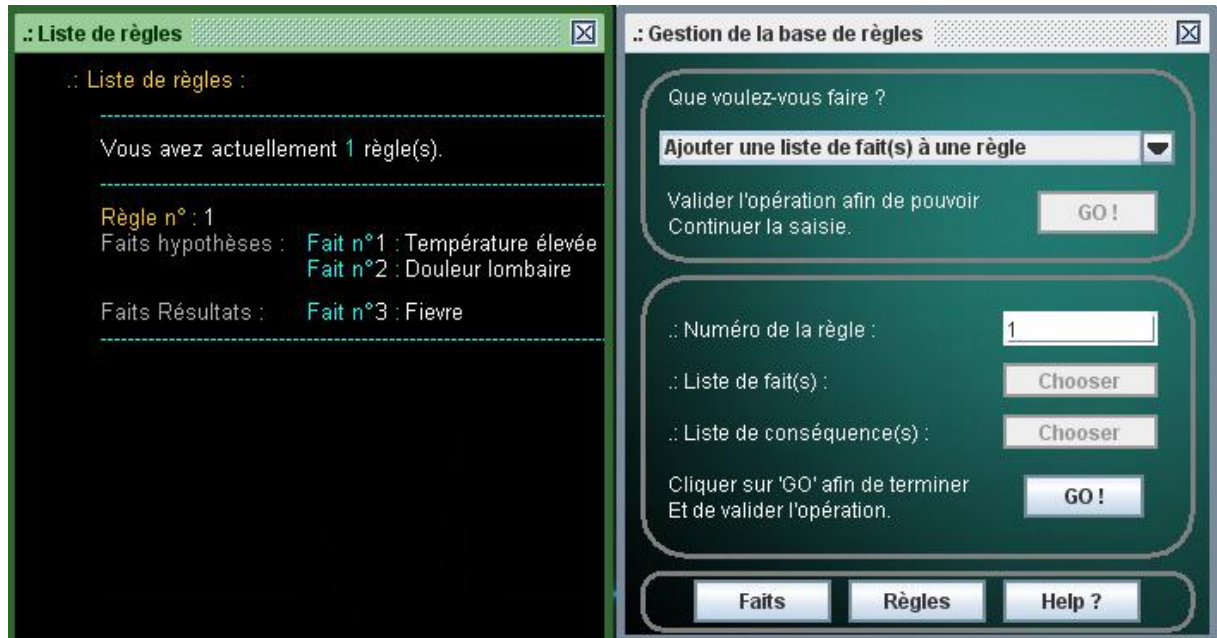


Figure 31 : Gestion des règles2

1.4. Gestion de recherches et inférences

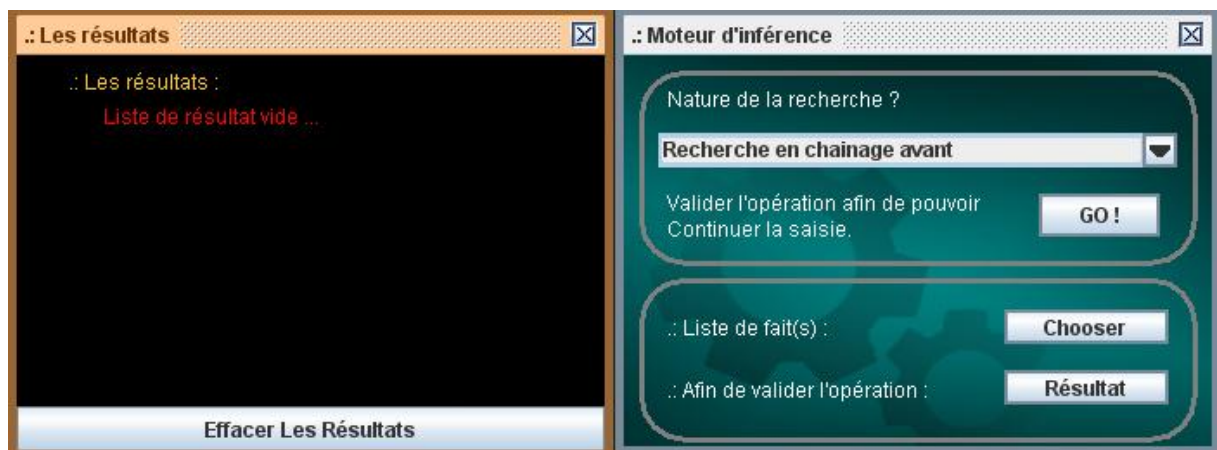


Figure 32 : Gestion de recherche1

Pour la gestion de la base des règles il y a deux menus :
 Un menu pour l'affichage des résultats et un autre pour faire la recherche.

La [Figure 32] montre l'état de la fenêtre avant le lancement de la recherche .

La [Figure 33] montre un exemple de recherche en chaînage avant et arrière avec les résultats.

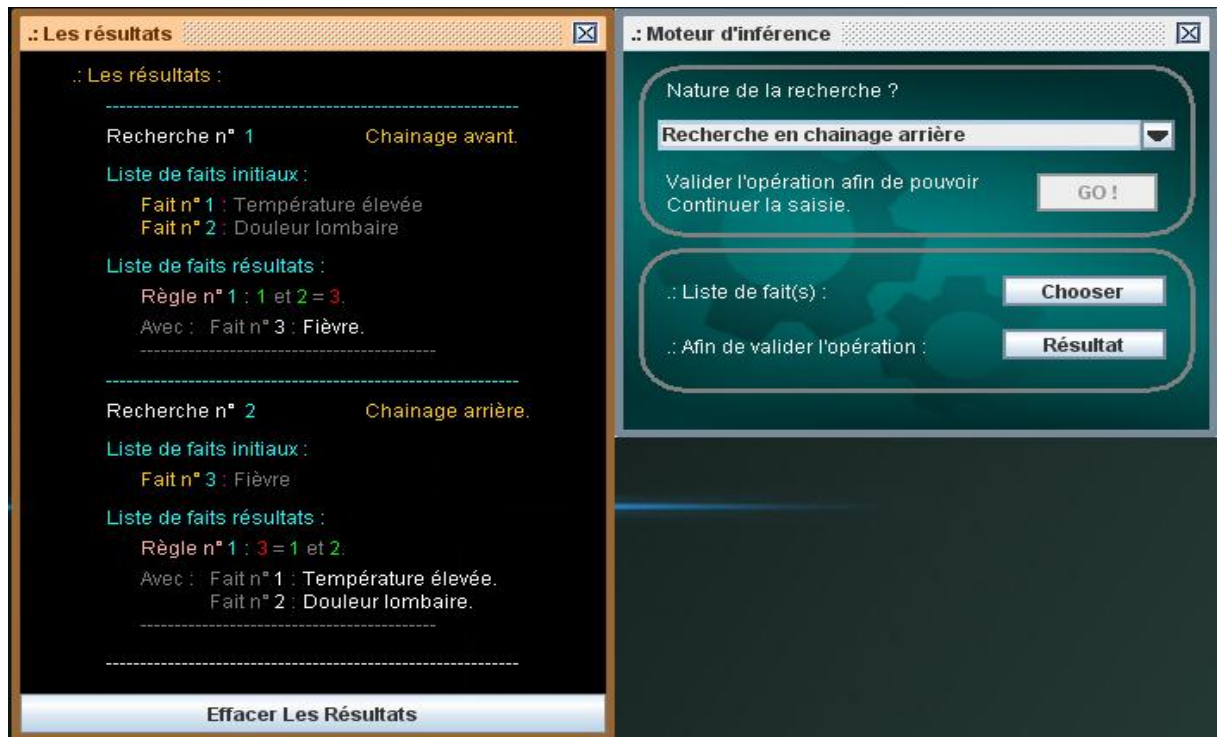


Figure 33 : Gestion de recherche2

1.5. A propos



Figure 34 : A propos de MyExpert

1.6. Help

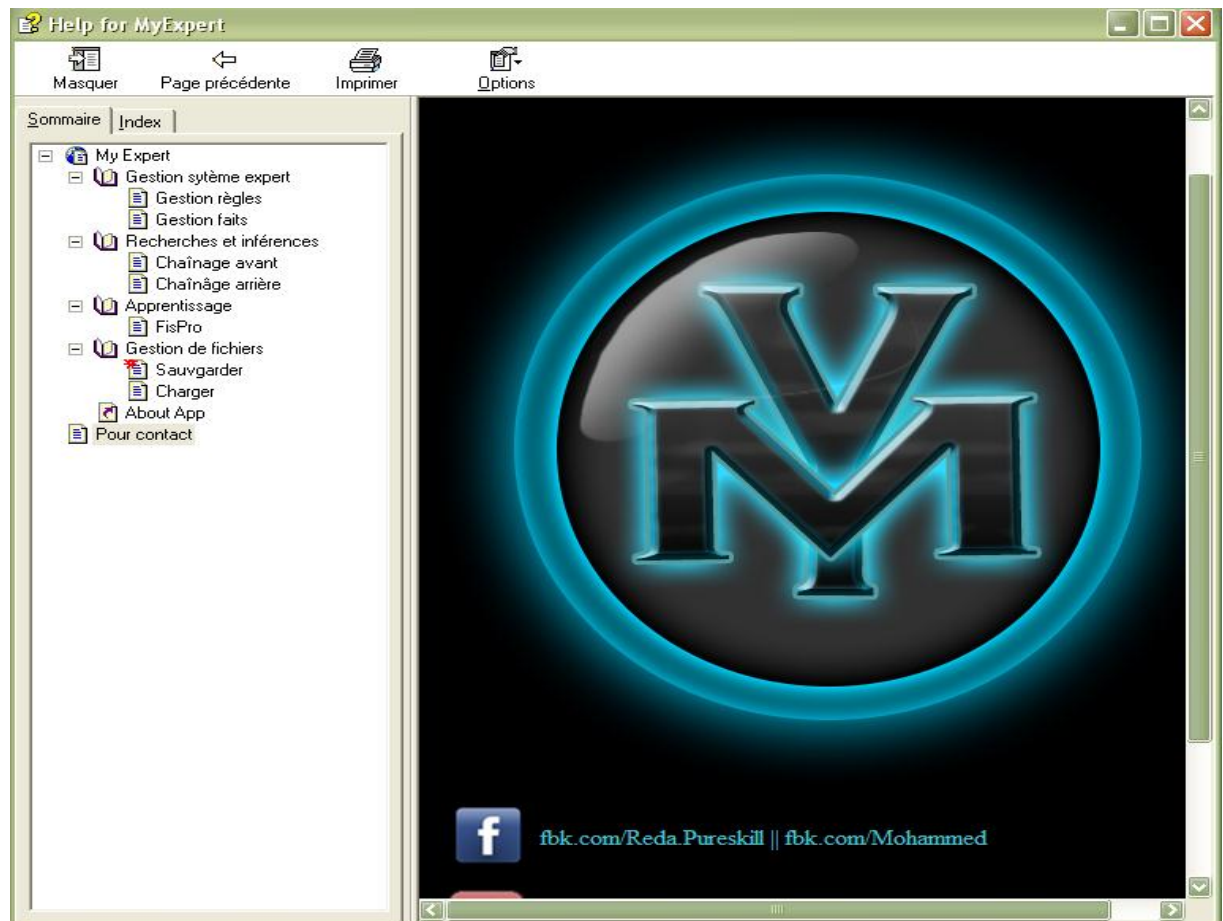


Figure 35 : Fichier help

Le fichier **help** montré dans la figure 35 facilite l'utilisation de l'application pour les débutants.

Il contient toutes les informations nécessaires pour la création et la gestion d'un système expert avec des exemples.

4. Conclusion

Dans ce chapitre nous avons décrit les outils utilisés pour la réalisation de l'application, ou on a montré l'AGL (Atelier génie logiciel) et les outils de création des ressources utilisées dans le design, ensuite les différents composants graphiques de l'application ont été expliqués.

CONCLUSION GENERALE

Conclusion générale et perspectives

Ce projet était bénéfique pour nous dans plusieurs sens. Il nous a permis, de nous perfectionner en améliorant nos connaissances en programmation et en conception et de bien comprendre et mettre en œuvre des systèmes experts et les systèmes d'aide à la décision en général dans le but de faciliter la prise de décision du médecin.

Dans la littérature, il existe plusieurs algorithmes d'aide à la décision médicale, mais le médecin a besoin toujours du cognitif et de l'informaticien pour construire ces systèmes

L'objectif principal de ce projet est de faciliter le plus que possible l'utilisation d'un outil d'aide à la décision et rendre le médecin autonome pour la création des nouvelles bases de connaissance.

Les systèmes d'apprentissage automatique constituent une voie nouvelle d'acquisition des connaissances de ce fait, on a relié notre application avec un outil d'apprentissage « FisPro » qui fait le traitement des bases de données externes, et capable de produire les règles d'inférences.

On a appliqué au maximum possible les règles de bases permettant d'avoir une application performante. Nous avons appliqué UML pour concevoir une grande partie de notre travail. Nous avons utilisé aussi Java pour implémenter notre application.

Dans la continuité directe de notre travail, nous avons jugé intéressant de proposer ces divers points de recherche pouvant faire l'objet d'étude à l'avenir :

- Utilisation des méthodes de sélection des variables pertinents avant la construction du système à partir de la base d'apprentissage.
- Enrichir le help avec des formations vidéo.
- Amélioration de l'application par ajout d'un système de gestion des dossiers de patients.

REFERENCES

- [1] Degoulet P, Fieschi M. Traitement de l'information médicale — Méthodes et applications hospitalières. Disponible sur google livres.
- [2] Abidi SSR. Healthcare knowledge management: The art of the possible. In: Knowledge Management for Health Care Procedures. Berlin: Springer, 2008;
- [3] Wright A, Sittig DF, Ash JS, Sharma S, Pang JE, Middleton B. Clinical decision support capabilities of commercially-available clinical information systems. JAMIA 2009;
- [4] Schnipper JL, Linder JA, Palchuk MB, Einbinder JS, Li Q, Postilnik A, Middleton B. "Smart forms" in an electronic medical record: Documentation-based clinical decision support to improve disease management. J Am Med Inform Assoc 2008;
- [5] Glaser J. Clinical decision support: The power behind the electronic health record. Healthc Financ Manage 2008;
- [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] Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications 2009
- [8] Stuart J. Russell et Peter Norvig, Artificial Intelligence : A Modern Approach, Upper Saddle River (New Jersey), Prentice Hall, 2009.
- [9] Grady Booch, James Rumbaugh, Ivar Jacobson (2000). Le guide de l'utilisateur UML.
- [10] The Birth of a killer application, 10 years photoshop, john knoll and Tomas knoll, February 2000.

Résumé

La complexité croissante des connaissances médicales sont devenues de plus en plus nombreuses au fil des années, De ce fait, une prise en charge des patients respectant les bonnes pratiques cliniques, nécessite que le médecin soit outillé pour réaliser ces tâches complexes.

Nous avons proposé une application pour la gestion efficace des système d'aide à la décision médicale « my_expert » qui facilite la tâche du médecin basé sur les modèles de l'intelligence artificiel. Ce système fiable et flexible d'aide au diagnostic en comparaison avec d'autre système existant

Abstract

The increasing complexity of medical knowledge have become increasingly numerous over the years, of this, a patient care following good clinical practices, requires that the doctor is equipped to perform these complex tasks.

We proposed an application for the effective management of medical decision support system called "my_expert" which facilitates doctor's task; it is based on artificial intelligence models. in comparison with other existing work, this system is reliable and flexible for diagnostic aid.

ملخص

التعقيد المتزايد للمعرفة الطبية أصبحت كثيرة بشكل متزايد على مر السنين، من هذا، ورعاية المرضى اتباع الممارسات الطبية الجيدة، يتطلب أن تم تجهيز الطبيب لأداء هذه المهام المعقدة. اقترحنا تطبيق فعال لإدارة أنظمة مساعدات القرارات الطبية "my_expert" لتسهيل مهمة الطبيب، هذا الأخير مصمم على أساس نماذج الذكاء الاصطناعي. بالمقارنة مع الأنظمة القائمة الأخرى، يعتبر هذا النظام موثوقا ومرنا للتشخيص الطبي.