

**République Algérienne Démocratique et Populaire**  
**Ministère de L'enseignement Supérieur et de la Recherche Scientifique**  
**Université de Hama Lakhdar El-Oued**  
**Faculté des Science et de Technologie**



***Rapport de fin d'étude***

**LICENCEACADEMIQUE**

**Domaine:** Mathématiques et Informatique

**Filière:** Informatique

**Spécialité:** Informatique Fondamentale

**Thème**

***Conception et Réalisation d'un système expert pour le  
diagnostic des maladies de tomate***

**Encadré par:**

***Mr.MEKLID Abdessalam***

**Présenté par:**

***Ghendir Mabrouk Soufiane***

***Dou Ahmed***

***Promotion 2015***

## *Résumé*

Objectif sommaire de la demande tout simplement dédié à fournir une assistance pour le diagnostic des maladies de tomates afin d'aider les agriculteurs à diagnostiquer les maladies.

Notre travail est la conception et la mise en œuvre des bases de production de base du système (d'experts) à l'aide de la base de connaissances; bases représentées par la logique propositionnelle l'aide d'un maladie table nommée dans le moteur de base de données et d'inférence 0 degré à réfléchir sur l'utilisation de la déduction pour induire un diagnostic basé sur les symptômes entrée par l'utilisateur.

### **Mots clés:**

Système Expert, Maladies de La Tomate.

### **ملخص**

الهدف من التطبيق ببساطة مكرس لتوفير المساعدة لتشخيص أمراض الطماطم من أجل مساعدة الفلاحين لتشخيص الأمراض.

عملنا يتمثل في تصميم وتنفيذ ( نظام خبير ) بأساس قواعد الإنتاج باستخدام قاعدة المعرفة؛ قواعدا يمثلها منطق اقتراحي باستخدام جدول اسمه مرض في قاعدة البيانات و محرك الاستدلال درجة • للتفكير باستخدام الاستنباط للحث على التشخيص انطلاقا من مدخلات الأعراض من قبل المستخدم.

لتحقيق هدفنا استخدمنا، منهجية UML للنمذجة والتصميم، ولغة البرمجة C # و SQL Server DBMS لإنجاز البرنامج.

### **الكلمات المفتاحية :**

النظام الخبير، أمراض الطماطم.

# Dedicace

Nous dédions notre projet à nos chers parents qui nous ont supportés

A nos chers frères et sœurs

A nos familles, amis et à nos enseignants

# Remerciements

Nous remercions dieu de nous avoir donné la force pour mener à bien ce projet.

Nous remercions nos parents pour leur soutien.

Nous remercions également nos frères et sœurs pour leur attention durant toute la période de la réalisation de ce travail.

Nous remercions notre frère **ABDESSALAM MEKLID** pour aide précieuse et conseils.

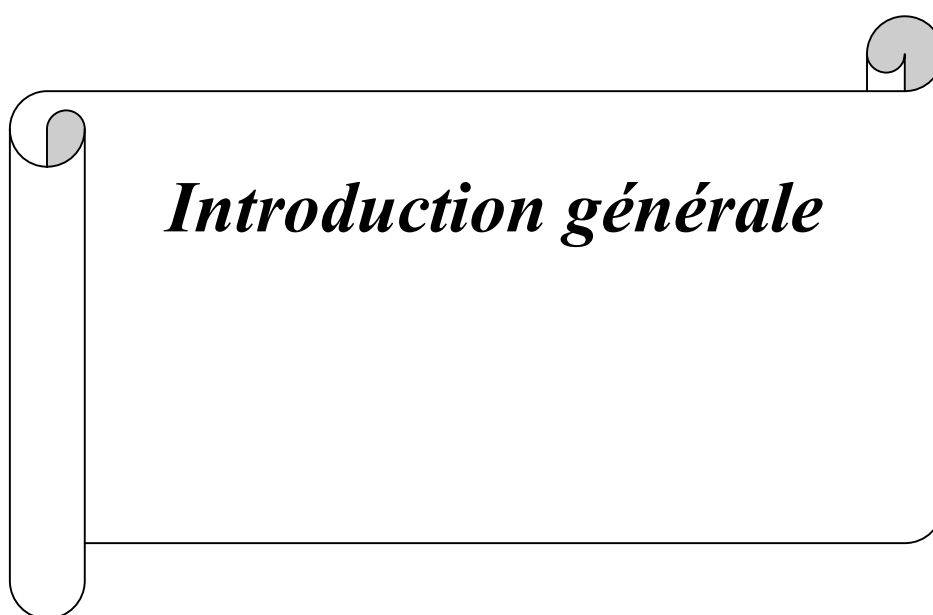
## Sommaire

<b><u>CHAPITRE I</u></b>	
<b>Introduction générale.....</b>	<b>2</b>
<b>I.1-Introduction.....</b>	<b>2</b>
<b>I.2- Motivation et Objectif :.....</b>	<b>2</b>
<b>I.3 Définition Système Expert .....</b>	<b>4</b>
❖ Définition 01: .....	4
❖ Définition 02:.....	5
<b>I.4-Le Rôle d'un système expert .....</b>	<b>5</b>
<b>I.5-L'architecture et les composants de base d'un système expert.....</b>	<b>6</b>
I.5.1-La base de Connaissance.....	7
I.5.2-Base des règles .....	7
I.5.3- Moteur d'inférence.....	8
I.5.3.1-Le cycle de base d'un moteur d'inférence.....	8
I.5.3.1.1- La phase d'évaluation .....	8
I.5.3.1.2- La phase d'exécution .....	8
<b>I.6- Le mode de Raisonnement... .....</b>	<b>10</b>
I.6.1- Le chaînage avant .....	10
I.6.2- Chaînage Arrière .....	11
I.6.3- Chaînage Mixte .....	11
<b>I.7.1- Les Avantages et Limites des Systèmes Experts.....</b>	<b>11</b>
I.7.1- Les Avantage.....	11
I.7.2- Les limites.....	12
<b>I .9- Conclusion.....</b>	<b>12</b>
<b><u>CHAPITRE II</u></b>	
<b>II.1- Introduction.....</b>	<b>14</b>
<b>II.2- Conception général.....</b>	<b>14</b>
<b>II.3- Modélisation.....</b>	<b>14</b>
II.3.1- Langage de conception( UML).....	15
II .3.2- Pourquoi nous utilisons UML?.....	15
<b>II .4 - Description du système.....</b>	<b>15</b>
<b>II .5- Diagramme de cas d'utilisation .....</b>	<b>16</b>
II .5.1-diagramme de cas d'utilisation de notre projet.....	16

❖ Le diagramme de cas d'utilisation notre Project:.....	17
II.5.1.1- Cas d'utilisation «Authentification» :.....	17
II.5.1.2- le diagramme cas d'utilisation « Diagnostiquer »:.....	18
II.5.1.3- Cas d'utilisation «ajouter Symptôme», «ajouter maladie» et «ajouter règle »	18
<b>II.5.2- Les diagrammes de séquence.....</b>	<b>18</b>
II.5.2.1- Le diagrammes de séquence Authentification :.....	18
II.5.2.2- Le diagrammes de séquence Diagnostic maladie :.....	19
<b>II.5.3- Diagramme de classe .....</b>	<b>20</b>
Diagramme de classe du projet .....	21
<b>Conclusion.....</b>	<b>21</b>
<b><u>CHAPITRE III</u></b>	
<b>III. 1- Introduction .....</b>	<b>23</b>
<b>III. 2- L'architecture de Notre Système Expert :.....</b>	<b>23</b>
III.2.1- Base de Connaissance :.....	23
III.2.2- Moteur d'inférence :.....	23
III.2.3- Interface :.....	25
<b>III.3- Description de l'environnement de développement :.....</b>	<b>25</b>
III.3.1- Microsoft Visual Studio :.....	25
<b>III.4- Les interface Authentification :.....</b>	<b>26</b>
III.4.1-Authentification .....	27
III.4.2 -Interface de Diagnostic des maladies :.....	28
<b>III.5- Déroulement du système :.....</b>	<b>28</b>
III.5.1- Exemple de déroulement Moteur d'inférence :.....	28
<b>Conclusion.....</b>	<b>32</b>
<b>Conclusion Générale.....</b>	<b>33</b>

**Liste des Figures**

Figure (I.1) Schéma illustratif d'un système expert .....	6
Figure (I.2) Cycle de base d'un moteur d'inférence.....	9
Figure (II .1) Schéma illustrative représente le fonctionnement général de notre SE...	14
La figure II.2 : Tableau indication des « acteurs » et « Use Case ».....	16
La figure II.3:Diagramme de cas d'utilisation.....	17
La figure II.4: diagrammes de séquence Authentification.....	19
La figure II.5: diagrammes de séquence Diagnostiquer .....	20
La figure II.11 : << les diagrammes de classe de notre projet>>.....	21
Figure III. 1 : << l'interface générale de Microsoft Visual studio>>.....	26
Figure III. 3.1 : Interface << Authentification >>.....	27
Figure III.3.2 : Interface << Authentification user >>.....	27
Figure III.3.3: << Interface de Diagnostic des maladies >>.....	28
Figure III.3.2<< Exemple de déroulement Moteur d'inférence cycle 1>>.....	29
Figure III.3.2<< Exemple de déroulement Moteur d'inférence cycle 2>>.....	29
Figure III.3.2<< Exemple de déroulement Moteur d'inférence cycle 3>>.....	30
Figure III.3.2<< Exemple de déroulement Moteur d'inférence cycle 4>>.....	30
Figure III.3.2<< Exemple de déroulement Moteur d'inférence cycle 5>>.....	31
Figure III.3.2<< Exemple de déroulement Moteur d'inférence cycle 6>>.....	31



## **Introduction générale**

Est ce que les machines peuvent raisonner ?

Pour répondre a cette question on va voir quel est le domaine qui s'intéresse de créer des telles machines intelligentes, ensuite on va voir les systèmes qui sont chargés de créer ces machines.

L'intelligence artificielle est une branche qui s'intéresse de créer des machines intelligentes simulant les processus de raisonnement et d'acquisition des connaissances de l'être humain, et cela dans les domaines de compréhension et de résolution des problèmes complexes.

Parmi les applications du secteur de l'intelligence artificielle les systèmes experts.

Ce sont des systèmes qui utilisent des connaissances pour répondre à une question ou la démontrer.

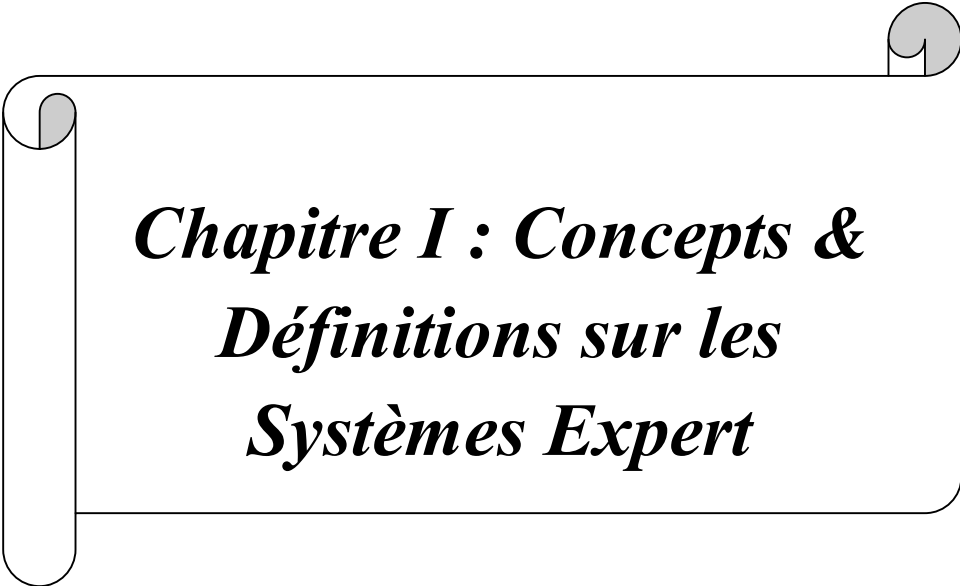
Dans ce rapport de fin d'études va tenter de créer une application pour un système expert qui fait le diagnostic des maladies de tomates « expertise dans le domaine agriculture ».

Notre travail consiste à concevoir et réaliser une application système expert a base des règles de production en utilisant une base de connaissance ; base des règles représenter par logique propositionnel sur une table appeler malade et un moteur d'inférence d'ordre 0 pour le raisonnement avec le chaînage arrière pour induire le diagnostic a partir les symptômes entrée par l'utilisateur.

Notre rapport est organisé suivant le plan décrit ci dessous : commençons par le chapitre (1) qui contient une introduction générale sur les systèmes experts et leurs composants de base ensuite les méthodes connues pour faire l'inférence.

Le chapitre (2) est consacré alla représentation delà conception et modélisation de notre système çà dire l'architecture qu'on a proposé et les méthodes qu'on a utilisé.

Le dernier chapitre consacré pour implémenter notre système expert.



***Chapitre I : Concepts &  
Définitions sur les  
Systèmes Expert***

## I.1 - Introduction

L'expert est une personne qui est capable de faire des choses dans un domaine particulier ne peut pas d'autres performances à la suite de son apprentissage et de l'expérience résultant de la pratique personnelle dans ce domaine. Et jusqu'à ce que ce système expert est de représenter l'expérience et les connaissances dont ils disposent et qui leur permettent d'agir efficacement et effectivement, le filtrage de ces expériences et de stockage et de maintenance et une utilisation accrue afin de ne pas rester pris au piège l'esprit et de nature à dissiper et la perte.

## I.2- Motivation et Objectif :

L'objectif de notre application consacrer tout simplement à offrir un niveau de soins des maladies des plantes de tomates plus amélioré pour les cultivateur au niveau de wilaya d'el oued , cela signifie obtenir des diagnostics plus précis et des options de traitement plus efficaces les agronome pourront systématiquement enrichir les procédures de diagnostic classique avec des nouvelles connaissances.

Le langage Médical est caractérisé par un vocabulaire extrêmement riche et difficile à manipuler, les Symptômes est les maladies sont nombreux en plus la largeur d'abréviation et d'acronyme gêner un peu d'ambiguïté (plusieurs termes désignant le même terme qui peut avoir plusieurs significations), donc il faut Formaliser ces information pour les mettre claires et précis, dans ce cas le langage est appelé un langage artificiel qui utilise des informations bien précis et bien formalisé .

## I.3- Définition Système Expert :

### ❖ Définition 01:

Un système expert est un logiciel qui reproduit le comportement d'un expert humain accomplissant une tâche intellectuelle dans un domaine précis.

Les systèmes experts sont nés d'une idée simple: rendre les ordinateurs intelligents en leur fournissant la connaissance des experts. Le fondement de concept de système expert réside dans la nature explicite de la connaissance.

En effet, celle-ci n'est absolument pas noyée dans un **ALGORITHME**, elle est formalisée par un ensemble des **règles** qui vont être **enchaînées** pour simuler le raisonnement d'un être Humain. [3]

Il existe dans la littérature plusieurs définitions d'un système expert :

**EDWARD FEIGNBAUM :**

« Les systèmes experts sont des programmes conçus pour raisonner habilement à propos de tâches dont on pense qu'elles requièrent une **expertise** humaine considérable ». [4]

**LOUIS E.FENZEL :**

Il est défini le (SE) comme un programme de IA incorporant une base de données et d'un moteur d'inférence. Le programme réagit comme un conseiller intelligent dans un domaine particulier.[4]

**DOMINIQUE PECCOUD :**

**Cite :** « un (SE) est un système logiciel informatique qui est capable de donner sur un sujet particulier un conseil égal en qualité celui qui pourrait être donné par un expert humain reconnu. »[4]

❖ **Définition 02:**

Les systèmes experts (SE) ne sont pas des outils et envisageables que pour des domaines possédant des experts humains, ces experts sont des individus qui connaissent bien le domaine à modéliser et sont aussi capables de transmettre leur savoir. ce dernier point est primordial.

Les systèmes experts sont utilisés lorsque les méthodes algorithmiques classiques ne peuvent pas être appliquées avec succès. Ils sont généralement conçus pour résoudre des problèmes de

Classification ou de décision (diagnostic médical , et le diagnostic des maladies des plantes , régulation d'échanges boursiers, etc.[4]

#### **I.4 - Le Rôle d'un système expert :**

Lorsque le processus intellectuel par lequel un humain évalue une situation ou prend une

Décision est **précisément modélisé**, il est relativement direct (attention : ce n'est pas un

Jugement de valeur !) De ce programme, c'est le cas par exemple, dans des domaines tels que la

Comptabilité, le calcul scientifique ou la commande numérique de machines-outils.

On ne considérera Pas (ni ici, ni dans la plupart des publications sur le sujet, mais pas toutes !) Que les logiciels résultant d'une telle programmation directe

D'un processus intellectuel parfaitement analysé et décrit pas à pas sont des (SE). Sinon tous les logiciels d'une certaine envergure pourraient être appelés

**Systèmes - experts** ce qui estomperait les caractères originaux et potentiellement féconds du concept.

Par contre, dans beaucoup de domaines socialement importants, tels que le diagnostic médical, l'orientation scolaire, la prospection minière l'organisation du travail, les spécialistes font largement appel à des connaissances éparses, parcellaires, souvent d'origine expérimentale ou de nature heuristique.[5]

### I.5- L'architecture et Les Composant de Base d'un Système Expert :

Un système expert peut se décomposer en deux composantes principales :

-La base de **connaissance**, elle-même composée d'une base de règles qui modélisent

La connaissance du domaine considéré et d'une base de faits qui contiennent les informations concernant le cas à traiter.

- Le moteur d'inférences dont le rôle est de raisonner à partir des données contenues dans la base de connaissances.

L'élément clé d'un système expert réside dans l'indépendance qui existe entre la base de connaissances et le moteur d'inférences. Elle permet une représentation des connaissances sous forme purement déclarative et explicite, c'est -à- dire sans lien avec la manière dont ces connaissances sont utilisées. L'avantage de cette d'architecture est qu'il est possible de faire évoluer les connaissances du système sans avoir à agir sur le mécanisme de raisonnement.

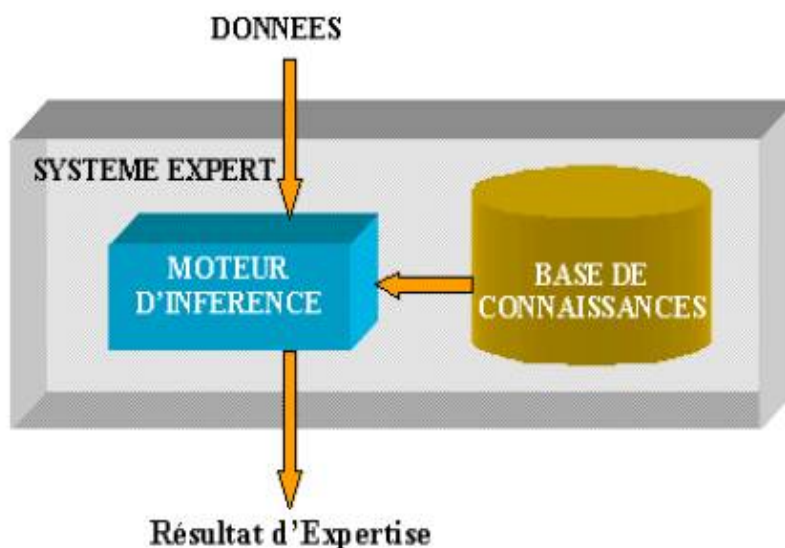


Figure I -1 : Schéma illustrative d'un système expert

### I.5.1- La base de Connaissance :

La base de connaissance contient toute l'information dont un expert humain aurait

Besoin pour s'acquitter de son travail, ceci dans un domaine donné.

Elle peut être **développée, modifiée, agrandie, transmise**, sans qu'il soit nécessaire de toucher au moteur d'inférence. [4]

La base de connaissances d'un système expert se décompose en deux parties. La première partie contient des faits spécifiques du domaine, les connaissances factuelles. On parle alors de base de **Faits** « **Connaissances assertion elles** ».

La seconde partie, la base de **Règles**, contient les règles qui vont permettre au système de raisonner à partir des faits. Ces connaissances déductives sont représentées par des règles appelées règles de production « **Connaissances opératoires** ».

### I.5.2 - Base des règles :

Des « **Connaissances opératoires** » ou « **Règles** » qui représentent le **savoir-faire** sur le

Domaine : elles indiquent quelles conséquences tirer ou quelques actions accomplir lorsque telle situation est **établie** « c'est-à-dire faits avérés » ou est **établir** « c'est-à-dire faits poursuivis ou hypothétiques ».

Une règle se présente sous la forme : **SI(X) ALORS(Y)**. X est appelé **prémisse** et Y conclusion. Alors que X est une conjonction de conditions, c'est-à-dire une suite de comparaisons d'attributs et de valeurs à l'aide d'opérateurs, la conclusion est une affectation , **Fait = Valeur**.

Grâce à ce type de représentation proche du langage naturel, la connaissance du système sera Facilement accessible à l'utilisateur qui pourra ainsi aisément la modifier ou l'agrandir.

De même que dans un système expert, la connaissance est traduite en règles, la méta- connaissance s'exprime par des métarègles (c'est-à-dire des règles sur la manière d'utiliser les règles).

L'organisation d'une base de connaissances au moyen de méta - règles reste essentiellement déclarative, contrairement à toute organisation basée sur une structuration a priori de l'ensemble des règles (écrire les règles dans un ordre donné, etc.).

### I.5.3- Moteur d'inférence :

Pour exploiter cette connaissance , le **module moteur d'inférence** est nécessaire pour relier la description d'un problème aux capacités d'analyse d'une situation donnée. De façon générale, le moteur d'inférences sera capable de répondre à des questions ou résoudre un problème, de raisonner et de tirer les conséquences impliquées par la connaissance incluse dans le système.

Son rôle principale est de simuler le raisonnement de l'expert humain en enchaînant les connaissances utiliser suivant une certaine logique, logique **d'ordre 0**, **d'ordre 0+** et **d'ordre1**.

#### Exemple :

**Ordre 0** : logique des propositions

*Si ((Ferrari) et (Michael)) Alors rapide*

**Ordre 0+** : logique des propositions typée (attribut - valeur)

*Si ((voiture = Ferrari) et (pilote = Michael))Alors vitesse=rapide*

**Ordre 1** : logique des prédicats

$\forall X, Y : \text{Si Voiture}(X) \text{ et Pilote}(X, Y) \text{ Alors Rapide}(X)$

**Le Fonctionnement du Module Moteur inférence** : il existe plusieurs types de chaînage

Le chaînage **Avant**, chaînage **Arrière**, chaînage **Mixte**.

#### I.5.3.1-Le cycle de base d'un moteur d'inférence:

Quelque soit le type ou la stratégie de contrôle utilisée, Le cycle de base d'un moteur

D'inférence comprend toujours les deux phases suivantes :

##### I.5.3.1.1- La phase d'évaluation:

Elle contient en principe **3** étapes : LA RESTRICTION, LE FILTRAGE, LA RESOLUTION DE CONFLITS.

##### I.5.3.1.2 La phase d'exécution: (SELECTION)

La première étape de la phase d'évolution appelée sélection (ou restriction) à partir d'un état

Présent ou passé (noté **BF** ici) de la base défauts st d'un état présent ou passé (noté **BR**) de la

Base des règles, un sous-ensemble **F1** de **BF** et un ensemble **R1**de **BR**qui, a priori, méritent d'être comparés lors de l'étape de **FILTRAGE** qui suit.[3]

La restriction peut alors consister à privilégier les faits- problèmes par rapport aux autres,

Ou même le problème le plus récemment apparu par rapport aux plus anciens. La distinction en familles de règles est souvent matérialisée par la définition de structures séparées.[3]

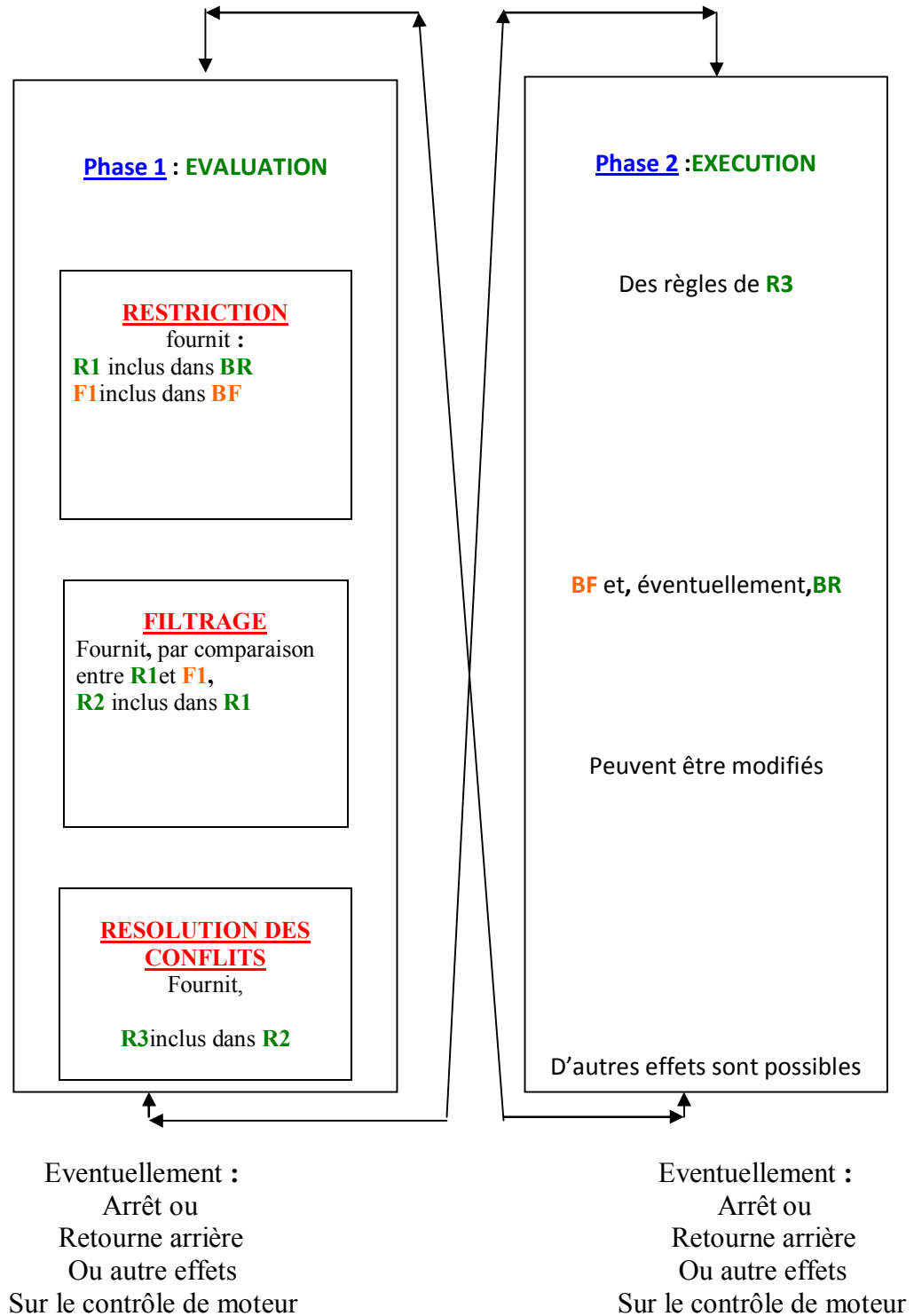


Figure I.2 : cycle de base d'un moteur d'inférence

Ce schéma est représentation de cycle de base d'un moteur d'inférence. **BR** et **BF** désignent respectivement les bases de règles et faits telles qu'elles sont au début de

**L'ÉVALUATION**. Le moteur commande l'enchaînement des étapes de la phase

**D'ÉVALUATION**, des phases **EVALUATION** et **EXECUTION**, puis des cycles complets

Entre eux. [3]

## I.6- Le mode de Raisonnement :

### I.6.1- Le chaînage avant :

**Le chaînage Avant permet de déduire les faits découlant de données initiales**

Le mécanisme du chaînage avant est simple : pour déduire un fait particulier, on déclenche les

Règles dont les prémisses sont connues jusqu'à ce que le fait à déduire soit également connu ou qu'aucune règle ne puisse être déclenchée.

On va analyser chaque fait et on va examiner toutes les règles où ce fait apparaît en prémisses. Pour les règles déclenchées, on va affecter les attributs en conclusion des valeurs qui leur correspondent. On dira que les faits ont été propagés. Ces attributs affectés feront partie du résultat final de l'expertise ; et, en même temps, ils seront eux-mêmes propagés. On fait cela jusqu'à l'épuisement des faits, et on communique les résultats à l'utilisateur.

```
ENTREES :    BF (base de faits),
              BR (bases de règles (R)),
              F  (fait que l'on cherche à établir)
DEBUT
  TANT QUE F n'est pas dans BF  ET
    QU'il existe dans BR une règle applicable
  FAIRE
    - choisir une règle applicable R (étape de résolution de
      conflits, utilisation d'heuristiques, de métarègles)
    - 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 DU TANT QUE
  SI F appartient à BF
  ALORS
    F est établi
  SINON
    F n'est pas établi
FIN
```

## **I.6.2-Chaînage Arrière :**

**Le chaînage Arrière permet d'atteindre des buts.**

Le mécanisme de chaînage arrière consiste à partir du fait que l'on souhaite établir, à rechercher toutes les règles qui concluent sur ce fait, à établir la liste des faits qu'il suffit de prouver pour qu'elles puissent se déclencher puis à appliquer récursivement le même mécanisme aux faits contenus dans ces listes. Le chaînage arrière est clairement un mécanisme d'induction : on vérifie les hypothèses en remontant depuis l'objectif. On cherche ainsi à vérifier si un fait est possible : celui-ci étant choisi selon des critères extérieurs.

## **I.6.3- Chaînage Mixte :(Chaînage Bidirectionnelle)**

**Le chaînage Mixte est un mélange des deux types de chaînages précédents.**

Le chaînage mixte utilise les deux chaînages présentés ci-dessus. On peut alors aussi bien raisonner à partir des faits que l'on connaît comme prédicats ou comme objectifs. Ce sont les

Caractéristiques du problème qui vont conditionner le chaînage qu'il est judicieux d'utiliser. Ainsi, lorsque les faits sont peu nombreux ou que le but est inconnu, il est préférable

D'employer un chaînage «**Avant**». Par contre, dans le cas où les buts sont peu nombreux ou précis, le chaînage «**Arrière**» est préconisé.[3]

## **I.7- Les Avantages et Limites des Systèmes Experts:**

N'importe qu'elle (SE) dans n'importe qu'elle domaine contient les avantages et les limites suivante :

### **I .7.1-Les Avantages :**

Le premier (SE) apparu pour le but de simuler ou actuellement remplacer la réflexion humaine donc c'est un système complètement intelligent par rapport aux programmes

Classiques, le (SE) caractériser par les avantages suivants :

-Ils disposent d'une interface utilisateur conviviale.

-Ils sont évolutifs, c'est -à- dire la base de connaissance n'est pas besoin d'être complète avant de commencer à utiliser le système.

-Ils sont capables d'expliquer le raisonnement qu'ils effectuent.

-Ils sont interactifs; en cas de données insuffisantes, l'utilisateur est invité à les Complété pour que le (SE) puisse poursuivre son raisonnement.[5]

### **I .7.2-Les limites :**

Nécessité de construire « à la main » l'ensemble de la base de connaissance et absence quasi totale d'apprentissage automatique.

- Formaliser ces connaissances et le raisonnement.
- Limitation des domaines d'expertise couverts et manque de connaissance du SE sur ses propres limites de compétence.
- Aspect superficiel des connaissances exploitées vs connaissance plus profonde sur laquelle l'expert humain appuie son raisonnement.

InterfaceLes meilleurs (SE) ne peuvent pas remplacer les spécialistes, même si les connaissances sont codées.[5]

## **I .9- Conclusion**

A partir de cette petite présentation générale des systèmes experts, nous avons choisi un système expert basé sur les règles de production qui utilise un moteur d'inférence d'ordre (0), C. Le chapitre 2 décrit en détail Notre conception.



***Chapitre II :***  
***Conception et***  
***Modélisation***

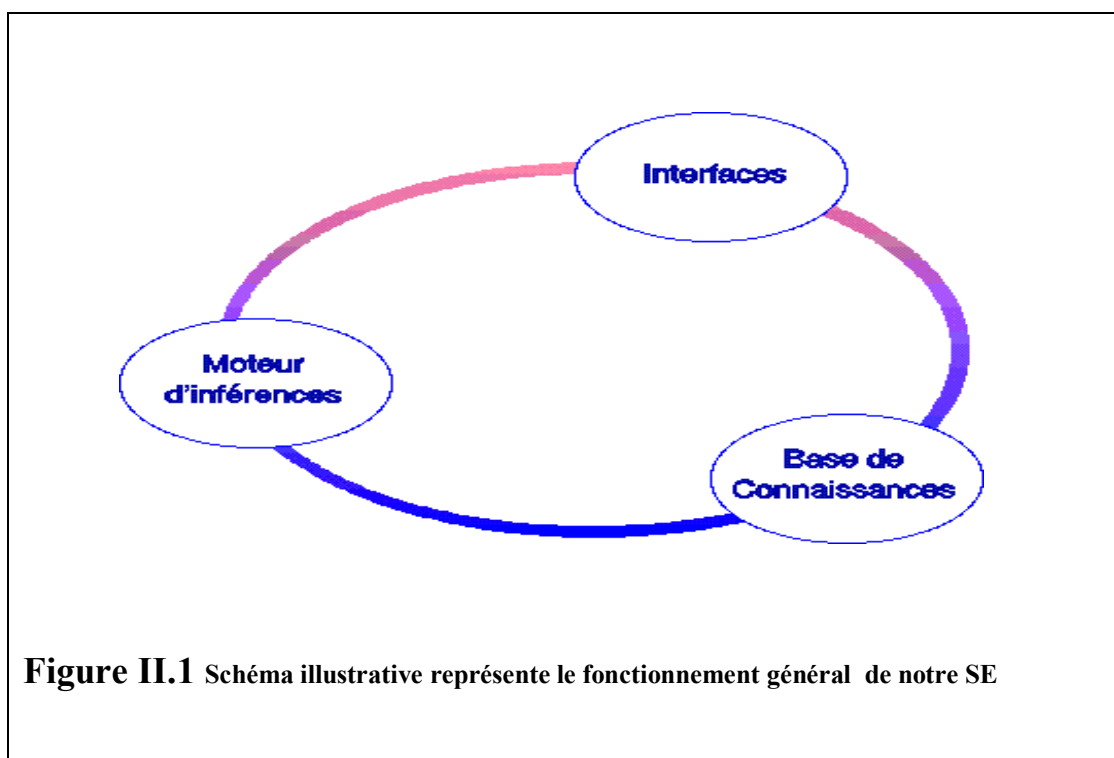
## II.1- Introduction :

D'après le chapitre précédent on a pris une vue générale sur les systèmes experts, leurs définition leurs architecture, en plus la définition d'une base de connaissance et le moteur d'inférence.

Dans ce chapitre en va voir la description de la conception générale de notre système, et l'algorithme d'inférence en chaînage arrière exploiter, en plus on va représenter notre modélisation en utilisant le différent diagramme d'UML.

## II.2- Conception général :

Celle-ci contient trois composantes interactives: une base de connaissances, un moteur d'inférences et une interface.



## II.3- Modélisation

Une bonne réalisation nécessite de faire une bonne modélisation. Pour ce faire cette modélisation, nous devons choisir une approche de modélisation adaptable au contexte de projet. Pour cela nous avons choisi l'UML. Dans ce chapitre, nous allons jeter un coup d'œil sur les origines du langage UML. Aussi, nous allons faire la modélisation de notre système en utilisant l'UML.

## II.3.1- Langage de conception UML:

Unified Modeling Language, qu'on a dire par " langage de modélisation unifié "

Est une notation permettant de modéliser un problème de façon standard. Ce langage est née la fusion de plusieurs méthodes 'existant au paravent, et devenu dé sur mais la référence en terme de modélisation objet, à untel point que sa connaissance est suivent nécessaire pour obtenir un poste de développeur objet. [8]

## II .3.2- Pour quoi nous utilisons UML?

De puis plus de 30 ans, la conception des bases de donnée ses réalisée à l'aide du

Modèle entité-association (Modèle de Chenet le MCD de Merise).

Ce modèle conceptuel fait ses preuves et la plupart des outils informatiques de modélisation

(Des tines principalement aux concepteurs français) l'utilisent en cor aujourd'hui.

La notation UML (Un fie de Modeling Langage ) s'est imposée de puisque l qu'es années pour la modélisation et le développement d'applications écrites dans un langage objet (C++ et Java principalement). Cette notation n'a pas été initialement pensée pour les bases des données mais elle permet d'offrir un même formalisme aux concepteurs d'objet se taux concepteurs de bases de données [8]

## II.4- Description du système:

Pour concevoir notre système, il fallut discuter avec l'expert dans les différents problèmes confronté pare au et collecter des différentes informations nécessaires permettant de répondre aux besoins de system .Notre étude dans ce chapitre est donc le fruit de cette discussion set de l'analyse des besoins. Nous va présenter quel que diagrammes UML,

Comme digramme de cas d'utilisation, de séquence et diagramme de classe. Aussi, nous allons fait le passage vers le modèle relationnel qui se compose de tables suivant de règles de passage bien définies.

## II .5- Diagramme de cas d'utilisation:

On utilise les diagrammes des cas d'utilisation pour représenter et structurer au niveau conceptuel, les besoins des utilisateurs et les objectifs correspondants du système [8].

Le but est d'identifier le structurer du domaine et leur sinter actions avec l'interface. Ce diagramme permet de déterminer le modèle objet sur le quelle système reposera.

### II .5.1- diagramme de cas d'utilisation de nôtre projet :

#### Elaboration du diagramme d'utilisation (use case)

Pour élaborer le diagramme de cas d'utilisation, commençons par identifier le s acteur de ce système. Il y en a deux acteurs principaux : Expert et l'utilisateur.

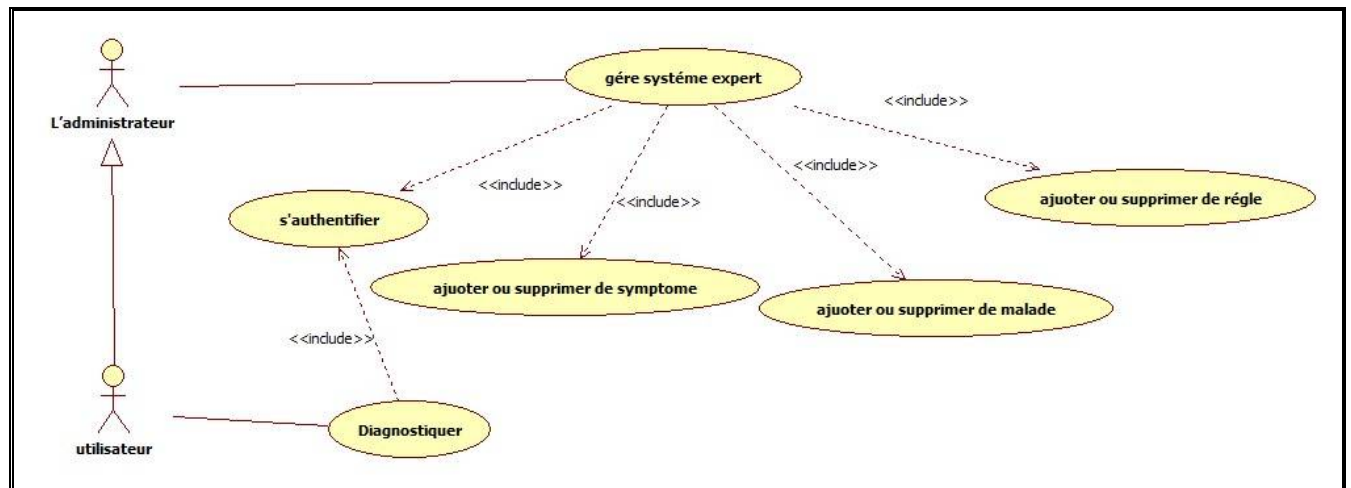
Ce tableau ci-dessous capicule l'ensemble des cas d'utilisation qui seront détaillés par la suite, en mettant en exergue les notions suivantes : Expert, diagramme de cas d'utilisation, les enchaînements.

<b>Cas d'utilisation</b>	<b>Acteur</b>
<b>Sélectionner Symptômes</b>	<b>Utilisateur</b>
<b>Ajouter Symptômes</b>	<b>Expert</b>
<b>Ajouter règles</b>	<b>Expert</b>
<b>Ajouter Maladies</b>	<b>Expert</b>

**La figure II.2 : Tableau Indication des « acteurs » et « Use Case »**

## ❖ Le diagramme de cas d'utilisation notre Project :

Est représenté par la figure suivante :



La figure II.3:Diagramme de cas d'utilisation

### II.5.1.1- Cas d'utilisation «Authentification» :

- **Intention** : Autorisation d'accès au système.
- Actions :
- Saisir le nom et le mot de passe.
- Valider puis autorisé.

#### Description de l'enchaînement:

#### Pré conditions :

Administrateur se connecté au système par le login.

#### Enchaînements :

**Enchaînement a** : demande d'accès au système. L'administrateur saisie le login pour Accéder au système.

**Enchaînement b** : l'acceptation d'accès.

Le système accepte login (le nom d'administrateur et le mot de passe).

### II.5.1.2 - le diagramme cas d'utilisation « Diagnostiquer »:

- ❖ Intention: fait le diagnostic d'une maladie et montré le résultat.
- ❖ Actions :
  - Peut aussi cliquer sur objet et affiche ses informations.
  - L'utilisateur peut sélectionner un objet dans toutes les couches.

### II.5.1.3- Cas d'utilisation «ajouter Symptôme», «ajouter maladie» et «ajouter règle »:

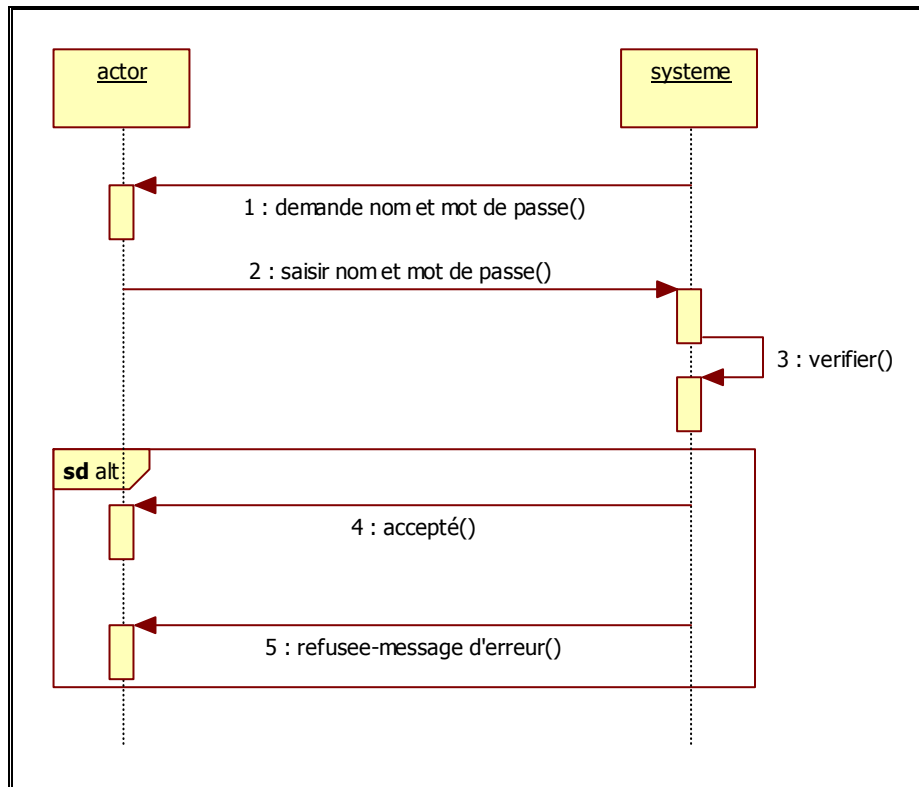
- Intention : ajout des symptômes, maladies ou règles aux tables de la base de données.
- Action :
  - ❖ Ajout des symptômes, maladies et règles aux tables de la base de données.

### II.5.2- Les diagrammes de séquence :

Le diagramme de séquences permet de cacher les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par le biais de messages.

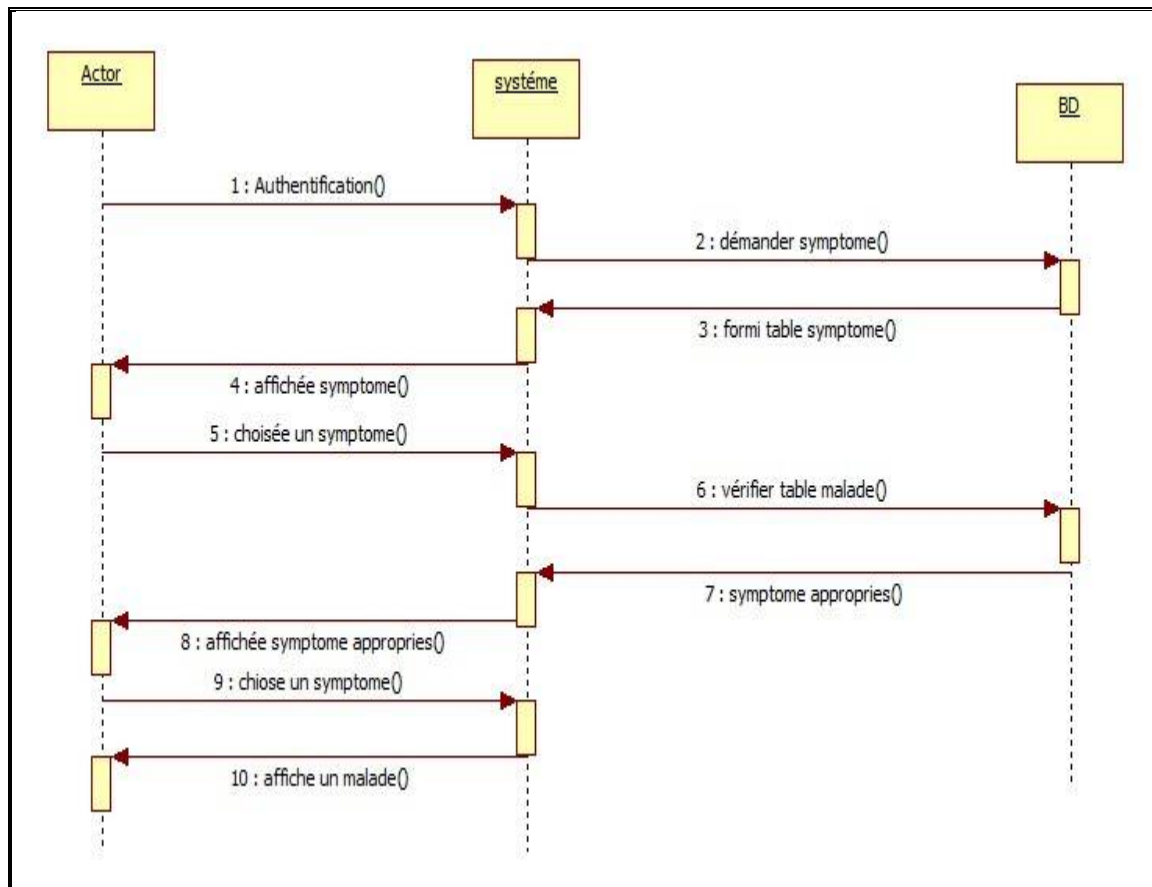
#### II.5.2.1- Le diagramme de séquence Authentification :



**La figure II.4: diagrammes de séquence Authentification**

### II.5.2.2-Le diagrammes de séquence Diagnostic maladie :

- 1) Entree au système (Authentification).
- 2) Le système lire les symptômes depuis la base de données.
- 3) Le système afficher les Symptômes au utilisateur.
- 4) L'utilisateur sélectionne un symptôme approprié.
- 5) Le processus de diagnostic ; si le but n'est pas atteindre fait le filtrage puis afficher les nouveaux Symptômes qui peut choisir pour atteint le but et retour a l'étape précédente 4.
- 6) Montrer à l'utilisateur la maladie.



**La figure II.5: diagrammes de séquence Diagnostiquer**

### II.5.3- Diagramme de classe :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

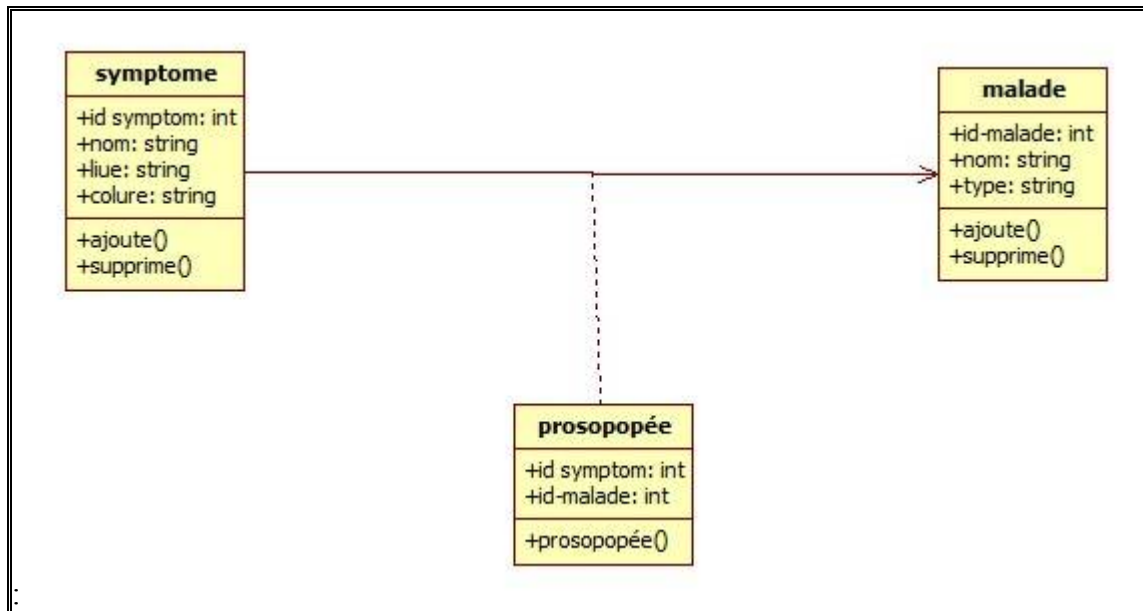
Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

## Conception et Modélisation

---

Elles sont finalement instanciées pour créer des objets (une classe est un moule à objet : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés)

### ❖ Diagrammes de classe de notre projet :



**La figure II.6 : << diagrammes de classe de notre projet >>**

### Conclusion:

Dans ce chapitre on a tracé les grandes bordures et la spécification de la conception de notre système. La modélisation, c'est une étape très importante qui précède le développement du système durant lequel nous avons suivi une démarche de développement, en passant par l'analyse et la conception du système.

Dans le chapitre suivant, nous allons aborder la réalisation de notre système.



***Chapitre III :***  
***Implémentation du système***

## **III. 1- Introduction :**

Ce chapitre présente les différents outils et techniques informatiques exploités pour la réalisation de notre application. En suite, on va présenter l'architecture de notre application avec un aperçu sur les maquettes des interfaces, ainsi que les différentes fonctionnalités de l'application illustrées par des figures pour expliquer comment notre application fonctionne.

## **III. 2- L'architecture de Notre Système Expert :**

L'architecture que l'on propose dans notre système expert a une base de règles de production composée de trois modules.

### **III.2.1- Base de Connaissance :**

C'est des données placées dans des tables de données : table symptôme et table maladie.

❖ **Symptôme** : représente les faits et les maladies.

❖ **Maladies** : représente les règles.

### **III.2.2- Moteur d'inférence :**

Représente le raisonneur ou le démonstrateur. On peut aussi dire que c'est le cœur du système. On a choisi comme stratégie de fonctionnement. On a choisi l'utilisation d'un moteur d'inférence avec chaînage avant.

### **Algorithme Chaînage Avant**

**Procédure chaînage\_avant()**

**Début**

premise q;

regle p;

Sélectionner les symptômes ;

p=tete\_r ; // la tête de la liste des maladies

**TQ (p≠NULL) faire**

**Si (p est Active) Alors**

**Si (la maladie p est activé par l'utilisateur) Alors**

            Copier le Résultat obtenue de p vers le variable Résultat

            Désactiver cette maladie p ;

            Ajouter tous les symptômes de p a la base des symptômes ;

            NBPA =NBP;

            Recommencer l'inférence ;

**Sinon**

            q=le pointeur vers la liste des symptômes de p ;

**TQ (q ≠ NULL) Faire**

**Si (q n'est pas Active) Alors**

**Si (ce symptôme est activer par l'utilisateur) Alors**

                        Activé le symptôme p ;

                        Incrémenter le NBPA ;

                        // Le nombre des symptômes active de cette règle

**Fin Si**

**Fin Si**

                q=le suivant symptôme;

**Fin TQ**

**Si (NBPA=NBP de maladie p) Alors**

                Copier le Résultat obtenue de p vers le variable Résultat

                Désactiver cette maladie p ;

                Recommencer l'inférence ;

**Fin Si**

**Fin Si**

        P=la suivante maladie p;

**Fin Si**

**Fin TQ**

**Si (le Résultat n'est pas vide) Alors**

    Afficher le Résultat ;

**Sinon**

    Afficher aucun Résultat ;

**Fin.**

## Implémentation

---

Cette structure représente la liste des maladies disponible dans notre base de connaissance tel que :

**NR** : représente le numéro de la règle (maladie).

**NBP** : représente le nombre des prémisses (symptômes) de cette maladie.

**NBPA** : représente le nombre des prémisses actives (symptômes disponible) de cette maladie.

**Active** : représente l'état de cette règle elle est applicable ou non.

**Prémisse** : représente un pointeur vers la liste des symptômes de cette maladie

**Conclusion** : représente la maladie lui-même.

### III.2.3- Interface :

C'est un moyen de communication entre notre système et l'utilisateur qui doit l'exploiter.

### III.3 -Description de l'environnement de développement :

Notre objectif est étendu en exploitant l'environnement de développement intégré Microsoft Visual Studio sous C#.

#### III.3.1- Microsoft Visual Studio :

Microsoft Visual Studio (en anglais: Microsoft Visual Studio) est un environnement de développement intégré majeur de Microsoft. Il permet GUI de programmation et de scripts ainsi que les sites de formulaire Web et Windows et les applications Web et les services Web pris en charge par Microsoft Windows et Windows Mobile et .NET Framework et Microsoft Silverlight.

Contenant atelier Vigiwal sur BB éditeur supporte Intel sens technologie et de réécrire le code, contient également un interprète révèle les erreurs d'exécution et l'interprète révèle des erreurs d'orthographe dans les codes contient également un modelleur de construire un concepteur graphique de l'interface utilisateur et Web designer catégories (classe (Informatique)) et a déterminé les règles du régime Les données et les rapports destinés à cristal.

# Implémentation

Dans ce travail, nous avons utilisé la version 2010 de, la figure suivante présente l'interface générale de :

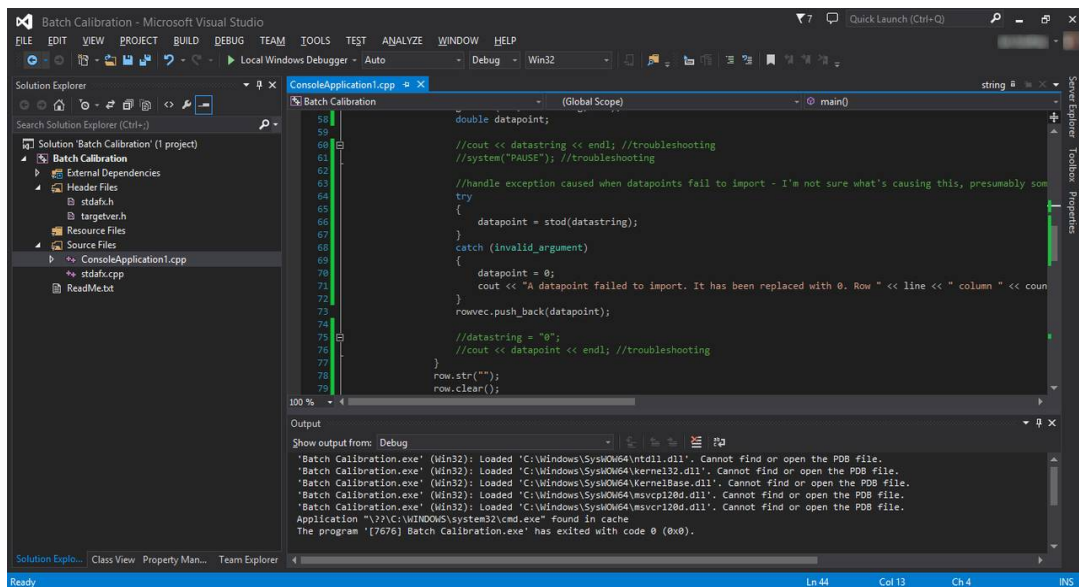
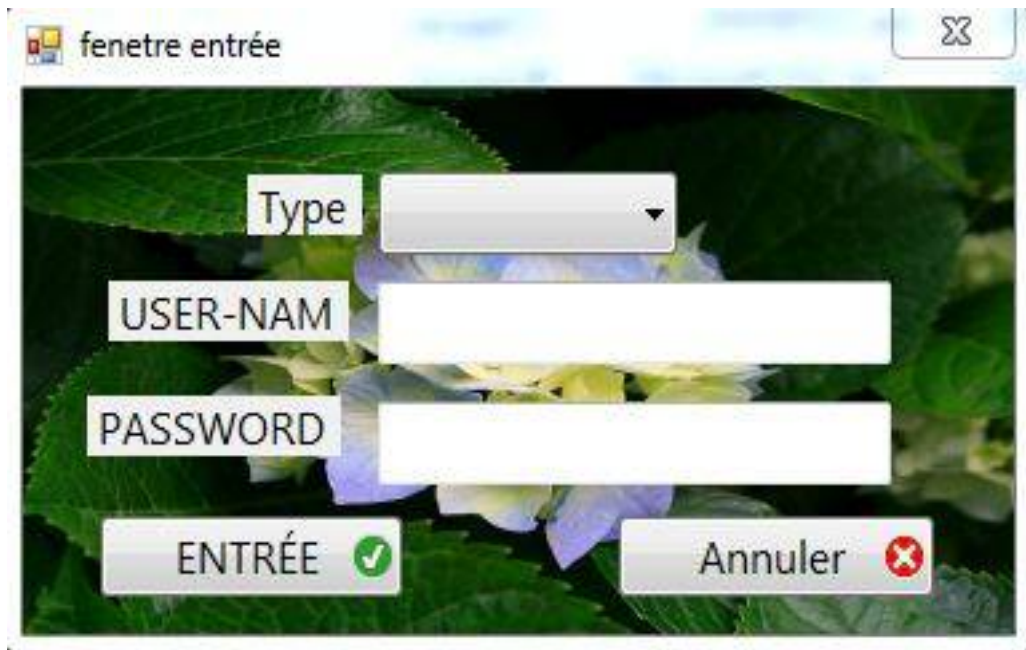


Figure III. 1 : l'interface générale de : Microsoft Visual studio

## III.4- Les interface Authentification :



**Figure III. 3.1 : Interface " Authentification "**

### III.4.1-Authentification

L'utilisateur peut seulement consulter la BD et sélectionner des symptômes.

La figure ci-dessous présente l'interface de l'Authentification :



**Figure III. 3.2: Interface " Authentification user "**

## III.4.2 -Interface de Diagnostic des maladies :



**Figure III.3.3 Interface de Diagnostic des maladies**

Cette l'interface présente l'essentiel de notre travail qui est le Diagnostic des maladies la figure affiche dans le cadre droite initialement tout les symptômes qui peut se produit une maladie et le cadre gauche présente tout les maladies qui peut existes dans notre système.

## III.5- Déroulement du système :

Après l'étape de l'authentification en mode utilisateur l'interface de diagnostic des maladies s'affiche.

1. l'utilisateur sélectionner des symptômes un par un en cliquant sur son nom dans le cadre droite de l'interface de diagnostic.

2. le moteur d'inférence sélectionne et affiche les maladies qui peut se produit par des symptômes sélectionné et filtré et afficher les nouveaux symptômes qui peut les choisis pour arrivé au but, retour a l'étape 1.si le but atteint passer a l'étape 3.

3. montrer le résultat et peut voir l'image de la maladie en cliquant sur son nom.

## III.5.1- Exemple de déroulement Moteur d'inférence :



Figure III.3.2 Exemple de déroulement Moteur d'inférence cycle 1

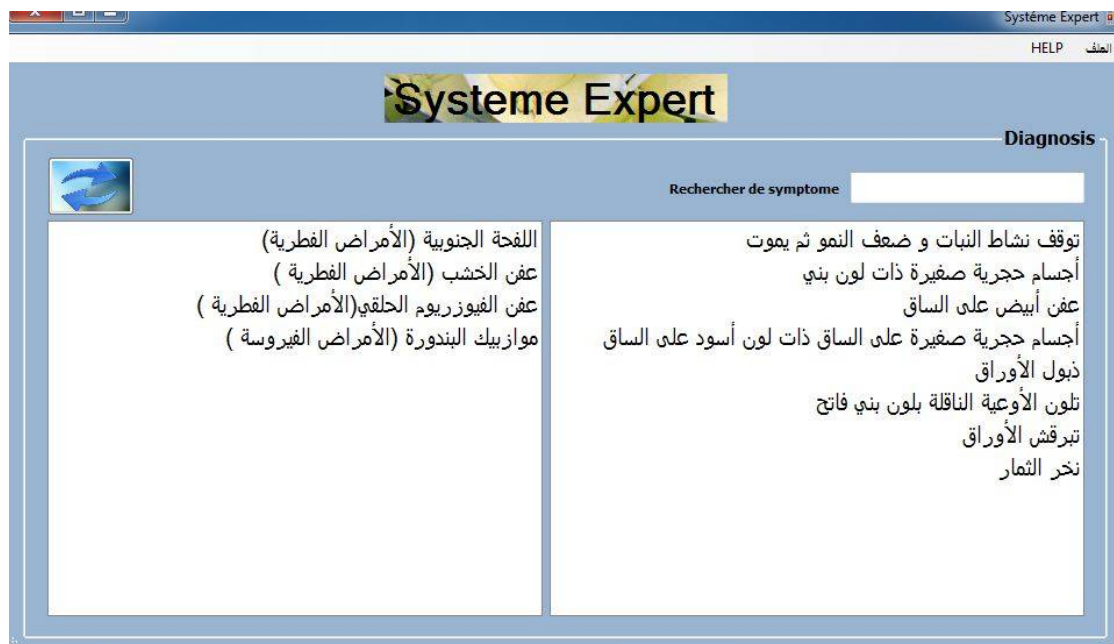


Figure III.3 Exemple de déroulement Moteur d'inférence cycle 2

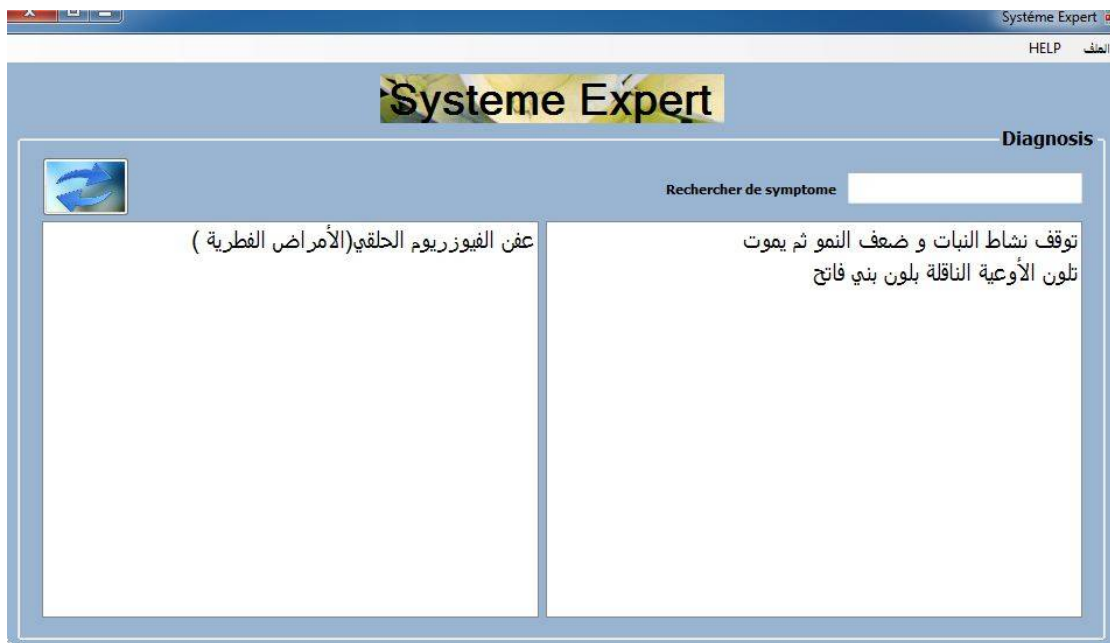


Figure III.3 Exemple de déroulement Moteur d'inférence cycle 3



Figure III.3 Exemple de déroulement Moteur d'inférence cycle 4



Figure III.3 Exemple de déroulement Moteur d'inférence cycle 5



Figure III.3 Exemple de déroulement Moteur d'inférence cycle 6

### **Conclusion**

Dans ce chapitre, nous avons expliqué comment appliquer l'information dans le programme avec l'utilisation des données et de la base du programme C # et l'objectif principal de ce répondre aux besoins de l'information des agriculteurs sur les maladies affectant le plant de tomate.

## **Conclusion Générale**

Comme conclusion de notre projet de fin d'étude , nous conçu un système expert dans le diagnostic des maladies de l'appareil de tomate basé sur les règle de production on utilisant le chaînage avant et le chaînage arrière , on réaliser un moteur d'inférence d'ordre 0.

Notre système n'est qu'une simulation d'un expert humain, il nous aidait à trouver des solutions approximatives on ne peut pas compter sur lui totalement puisqu'il existe plusieurs variations qu'il ne pas contenir

Donc le system expert ne peuvent pas remplacer l'expert humain dans n'importe quel domaine.

Comme perspective on augmenter l'ordre de notre moteur à l'ordre de notre moteur à l'ordre 1 en outre on va tester notre application dans différents domaines.

## Bibliographie

[1]: كتاب شامل عن الطماطم زراعتها أمراضها, تأليف و نشر: م-ز/ احمد مروان محمد احمد ١١-05-٢٠٠٨.

[2]: كتاب : « أمراض البندرة » دليل عملي لمراحل نمو و إنتاج البندورة

من [Tomato Diseases] A Practical Guide for Seedsmen, Growers and Agricultural devisors  
إعداد: المهندس الزراعي عيسى جرجسو المهندسة الزراعية مها عيسمن دار الدراسة للحماية البيئة سورية.

[3] : *Les systèmes experts* , Gilles Balmisse ,septembre 2002.

[4] : *Les systèmes experts principes et exemple*, Henri FARRENY.

[5] : Mémoire : Conception et réalisation d'un système expert basé sur les règles de production Application médicale ,Necib K, Lebcir B et Lehouimel F ; juin 2005 Biskra.

[6]: Site web: <http://www.tomatodirt.com/tomato-books.html>

[7]: Site web: [http://www.meleigi.com/news.php?action=list&cat\\_id=33](http://www.meleigi.com/news.php?action=list&cat_id=33)

[8] : Cours UML, Bali Ahmed 2013Eloued.