

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'INFORMATIQUE

## **Mémoire de Fin d'Etudes de MASTER ACADEMIQUE**

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Systèmes Informatiques**

*Présenté par*

**BELKACEMI Omar**

**DJOUZI Kheyreddine**

Thème

# **Conception et réalisation d'un système expert pour l'évaluation des dommages post-sismiques.**

*Mémoire soutenu publiquement le 28/09/2017 devant le jury composé de :*

**Président : M Rachid AHMED OUAMER**

**Encadreur : M Mohammed RAMDANE**

**Co-Encadreur : M Karim AKKOUCHE**

**Examineur : M Mohand HAMIZI**

## *Remerciements*

*En premier lieu nous remercions le bon dieu le tout puissant de nous avoir donné le courage, la patience, la force et la volonté pour réaliser ce mémoire.*

*Nous tenons à remercier Monsieur M.RAMDANE, enseignant à l'UMMTO, pour avoir accepté de diriger notre travail. Son soutien, ses conseils et sa disponibilité nous ont été d'un grand apport dans l'aboutissement de notre travail. Qu'il veuille bien trouver ici l'expression de notre gratitude et notre profond respect.*

*Nous remercions vivement messieurs M.DAOUI, N.HANNACHI, M.HAMIZI et KAKKOUCHE pour s'être dévoués à nous transmettre leur savoir et pour avoir mis à notre disposition tous les documents dont nous avons besoin. Nous les remercions aussi pour leur extrême gentillesse et disponibilité.*

*Nous remercions tous les enseignants du département informatique de la faculté de Génie Electrique et Informatique de l'université MOULOUD MAMMERI pour leur disponibilité, leurs conseils tout au long de notre cursus au sein de ce département.*

*Que les membres du jury trouvent ici nos remerciements les plus vifs pour avoir accepté d'honorer par leur jugement notre travail.*

*Nos sincères sentiments vont à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce projet. En particulier nos chères familles et nos amis(es).*

## *Dédicace*

*Je dédie ce modeste travail à :*

*Mes très chers parents,*

*Qui m'ont toujours fait confiance et n'ont jamais cessé*

*De m'encourager et me soutenir,*

*Ma chère sœur,*

*Toute ma famille sans exception,*

*Mon collègue et sa famille,*

*Mes amis et mes camarades.*

***KHEYREDDINE***

## *Dédicace*

*Je dédie ce modeste travail à :*

*Mes très chers parents qui m'ont énormément soutenu et encouragé tout au long  
de mon cursus,*

*Mes adorables sœurs LAMIA, FATMA et NADIA, qui m'ont toujours soutenu,*

*Mes chers frères AHMED et SOFIANE qui étaient toujours à mes côtés,*

*Tous mes amis.*

*OMAR*

# *Sommaire*

---

# SOMMAIRE

<i>INTRODUCTION GÉNÉRALE</i> .....	1
------------------------------------	---

## *Chapitre I : GÉNÉRALITÉS SUR LE DOMAINE D'EXPERTISE*

Introduction : .....	3
I. Généralités sur le domaine d'application : .....	3
I.1. Éléments structuraux : .....	3
I.1.A. Élément principale: .....	3
I.1.B. Élément secondaire: .....	3
I.2. Éléments non structuraux : .....	3
I.3. Système de contreventement : .....	3
I.4. Mur de contreventement : .....	3
I.5. Voile de contreventement : .....	3
I.6. Système porteur : .....	3
I.7. Construction en Portiques : .....	4
I.8. Construction en maçonnerie : .....	4
I.9. Règlement parasismique algérien (RPA) : .....	4
II. Bâtiments : .....	4
II.1. Définition : .....	4
II.2. Typologies des bâtiments : .....	4
II.3. Composants des bâtiments : .....	5
II.3.A. Bâtiments en maçonnerie : .....	5
II.3.B. Bâtiments en portique : .....	6
III. Tremblement de terre (séisme): .....	7
III.1. Définition : .....	7
III.2. Paramètres d'un séisme : .....	7
III.2.A. Foyer d'un séisme : .....	7
III.2.B. Epicentre : .....	7
III.2.C. Intensité : .....	7

III.2.D. Accélération d'un séisme :	7
III.2.E. Faille :	7
III.2.F. Ondes sismiques :	7
III.2.G. La magnitude :	7
III.2.H. Classification des séismes :	7
III.3. Les séismes en Algérie :	8
III.4. Zonage sismique :	10
IV. Moyens utilisés pour l'évaluation les dommages post sismique :	11
V. Évaluation des dommages Globaux :	14
V.1. Orientation sur l'évaluation des dommages causés par le séisme :	15
V.2. Procédure d'inspection des dommages causés par les séismes :	15
V.2.A. Inspection détaillée des dommages sismiques :	15
V.2.B. Inspection suivant la fiche d'évaluation du CTC ou CGS :	17
VI. Conclusion :	18

## *Chapitre II : GÉNÉRALITÉS SUR LES SYSTÈMES EXPERTS*

Introduction :	19
I. Généralités sur les systèmes experts :	19
I.1. Définition d'un système expert :	19
I.1.A. Définition 1 :	19
I.1.B. Définition 2 :	19
I.1.C. Définition 3 :	19
I.2. Historique des systèmes experts :	19
I.3. Caractéristiques des systèmes experts :	20
I.3.A. Le langage de représentation de la connaissance :	20
I.3.B. Posséder les facultés de :	20
I.4. Architecture des systèmes experts :	21
I.4.A. La base des connaissances (BC) :	21
I.4.B. Le moteur d'inférence (MI):	22
I.4.C. Les interfaces :	22
I.4.D. Le module d'acquisition des connaissances :	22
I.4.E. Le module d'explication :	22

I.5.	Principe des systèmes experts :	24
I.6.	Typologie des systèmes experts :	24
I.6.A.	Systèmes expert dédié :	24
I.6.B.	Générateurs de systèmes experts (Shell) :	24
I.7.	Domaines d'application des systèmes experts :	24
I.8.	Objectifs des systèmes experts :	25
I.9.	Avantages et inconvénients des systèmes experts :	25
II.	Représentation des connaissances :	26
II.1.	Représentation logique :	26
II.1.A.	Logique d'ordre 0 (logique propositionnelle) :	26
II.1.B.	La logique d'ordre 1 (logique des prédicats) :	27
II.1.C.	La logique d'ordre 1 <sup>-</sup> :	28
II.1.D.	La logique d'ordre 0 <sup>+</sup> :	28
II.1.E.	La logique de l'incertain :	28
II.2.	Règles de productions :	29
II.3.	Réseaux sémantiques :	30
II.4.	Représentation orienté objet :	32
II.4.A.	Langage de classes :	32
II.4.B.	Langage de frames et de scripts :	34
II.4.C.	Langage d'acteurs :	37
III.	Principes du fonctionnement d'un moteur d'inférence dans un système de production :	37
III.1.	Le cycle de base d'un moteur d'inférence :	37
III.1.A.	La phase d'évaluation :	38
III.1.B.	La phase d'exécution :	38
III.2.	Le mode d'invocation des règles :	39
III.2.A.	Le chaînage avant :	40
III.2.B.	Le chaînage arrière :	40
III.2.C.	Le chaînage mixte :	40
III.3.	La stratégie de recherche :	40
III.3.A.	La recherche en largeur d'abord :	40
III.3.B.	La recherche en profondeur d'abord :	41
III.3.C.	La recherche en profondeur limitée :	42
III.3.D.	Les recherches heuristiques :	43

III.4. La stratégie de contrôle : .....	44
III.4.A. La monotonie : .....	44
III.4.B. La non monotonie : .....	44
III.4.C. L'hypothèse du monde clos et du monde ouvert : .....	44
III.4.D. Le régime de contrôle par tentative : .....	44
IV. Conclusion : .....	44

### *Chapitre III : ANALYSE ET CONCEPTION*

Introduction : .....	46
I. Analyse : .....	46
I.1. Choix du domaine et objectif du système : .....	46
I.2. Intérêts du système EDPS : .....	46
I.3. Les utilisateurs de notre système : .....	47
I.3.A. Les utilisateurs ordinaires : .....	47
I.3.B. Les administrateurs : .....	47
I.4. Les intervenants du projet : .....	47
I.5. Extraction de la connaissance : .....	47
II. Conception : .....	48
II.1. Conception globale : .....	48
II.1.A. La base de connaissances (BC) : .....	48
II.1.B. Le moteur d'inférence (MI) : .....	49
II.1.C. Le module d'acquisition des connaissances (MAC) : .....	49
II.1.D. Les modules d'interfaces : .....	49
II.1.E. Le module d'aide (MA) : .....	50
II.1.F. Le module d'explication (ME) : .....	50
II.2. Conception détaillée : .....	52
II.2.A. La conception logique : .....	52
II.2.B. La conception structurelle : .....	55
III. Conclusion : .....	72

## Chapitre IV : RÉALISATION

Introduction :	73
I. Description des outils de travail :	73
I.1. Aspect matériel :	73
I.2. Aspect logiciel :	73
I.2.A. Le système d'exploitation :	73
I.2.B. Environnement de développement :	73
I.3. Prolog :	74
I.3.A. Définition :	74
I.3.B. SWI-Prolog :	74
I.4. Java :	75
I.4.A. Définition :	75
I.4.B. Historique :	76
I.4.C. Qu'est ce que le JDK ? :	76
I.4.D. Les caractéristiques du langage Java :	77
I.4.E. Java et les bases de données :	78
I.4.F. Présentation de l'EDI NetBeans :	79
I.5. Interfaçage entre SWI-Prolog et Java :	80
I.5.A. Qu'est ce que la JPL :	80
I.5.B. Présentation générale de quelques classes utilisées de la JPL :	82
I.6. MySQL :	83
I.6.A. Définition :	83
I.6.B. WAMP :	83
I.6.C. PHPMyAdmin :	83
II. Présentation des interfaces du logiciel :	84
II.1. L'interface d'accueil :	84
II.2. L'interface d'authentification pour l'expert :	85
II.3. L'interface Expert :	86
II.4. L'interface Recherche :	86
III. Les tests :	87
IV. Conclusion :	90
<i>CONCLUSION GÉNÉRALE</i> :	91

<i>BIBLIOGRAPHIE</i> .....	93
<i>WEBOGRAPHIE</i> .....	95

### *Annexe A*

I. Définition de l'UML : .....	96
II. Un peu d'histoire : .....	96
III. Présentation générale des diagrammes : .....	97
III.1. Les diagrammes structurels ou statiques : .....	97
III.2. Les diagrammes de comportement : .....	97
IV. Les diagrammes utilisés pour la conception de notre système : .....	98
IV.1. Diagramme de cas d'utilisation : .....	98
IV.1.A. Définition : .....	98
IV.1.B. Éléments de modélisation des diagrammes de cas d'utilisation : .....	98
IV.2. Diagramme de séquence : .....	99
IV.2.A. Définition : .....	99
IV.2.B. Éléments de modélisation des diagrammes de séquences : .....	99
IV.3. Diagramme de classe : .....	100
IV.3.A. Définition : .....	100
IV.3.B. Les notions utilisées par le diagramme de classes : .....	100

### *Annexe B*

## *LISTE DES FIGURES*

Figure 1: Ensemble des bâtiments.....	4
Figure 2 : Représentation des composants du bâtiment en maçonnerie.....	5
Figure 3 : Représentation des composants d'un bâtiment en poteaux-poutres. ....	6
Figure 4 : Schéma illustrant les paramètres d'un séisme. ....	8
Figure 5 : Zonage sismique du territoire national établi par le RPA 99 V2003.....	10
Figure 6 : Le modèle du formulaire d'évaluation utilisé par les autorités algériennes (Partie 1). ....	12
Figure 7 : Le modèle du formulaire d'évaluation utilisé par les autorités algériennes (Partie 2). ....	13
Figure 8 : Schéma de référence pour les fissures dans la maçonnerie. ....	15
Figure 9 : Catégorie de dommage pour les constructions en maçonnerie et en Béton Armé. ....	17
Figure 10: Architecture de base d'un système expert. ....	23
Figure 11: Schéma simplifié illustrant le principe des systèmes experts. ....	24
Figure 12: Réseau sémantique simple. ....	30
Figure 13: Représentation d'une requête par réseau sémantique. ....	31
Figure 14: Classe d'objets et objet. ....	34
Figure 15: Cycle de base d'un moteur d'inférence. ....	39
Figure 16: Recherche en largeur d'abord. ....	41
Figure 17: Recherche en profondeur d'abord.....	42
Figure 18: Recherche en profondeur limitée. ....	43
Figure 19: Architecture détaillée du système EDPS. ....	51
Figure 20: Le diagramme de contexte. ....	56
Figure 21: Diagramme de cas d'utilisation global du système EDPS.....	59
Figure 22: Diagramme de séquence de cas d'utilisation «S'authentifier en tant qu'administrateur ». ....	64
Figure 23: Diagramme de séquence de cas d'utilisation « consulter une évaluation ».....	65
Figure 24: Diagramme de séquence de cas d'utilisation « ajouter un expert ». ....	67
Figure 25: Diagramme de séquence de cas d'utilisation « évaluer une construction en tant qu'expert ». ....	69
Figure 26: Diagramme de classe du système EDPS.....	71
Figure 27: L'IDE SWI-Prolog.....	75
Figure 28: L'interface de l'IDE NetBeans 8.2. ....	80
Figure 29: Architecture des classes de la JPL. ....	81
Figure 30: Schéma de la JPL.....	81
Figure 31: L'interface PHPMyAdmin.....	84

Figure 32: Interface d'accueil du logiciel. ....	85
Figure 33: Interface d'authentification de l'expert. ....	85
Figure 34: L'interface Expert. ....	86
Figure 35: Interface de recherche. ....	86
Figure 36: Formulaire de renseignements généraux sur la construction. ....	87
Figure 37: Formulaire problèmes du sol et constructions adjacentes. ....	87
Figure 38: Interface "Fondations et infrastructure". ....	88
Figure 39: Interface "Structure résistante". ....	88
Figure 40: Interface "Structure secondaire". ....	89
Figure 41: Interface "Résultat". ....	89
Figure 42: Exemple de diagramme de cas d'utilisation. ....	99
Figure 43: Exemple de diagramme de séquence. ....	100
Figure 44: Représentation graphique d'une classe. ....	101
Figure 45: Relation d'association. ....	102
Figure 46: Relation de généralisation. ....	102
Figure 47: Relation de composition. ....	103

## *LISTE DES TABLEAUX*

Tableau I.1 : Éléments d'un bâtiment en maçonnerie. ....	5
Tableau I.2 : Éléments d'un bâtiment en portique. ....	6
Tableau I.3 : Historique des séismes en Algérie. ....	9
Tableau I.4 : Échelle d'évaluation de degré de dommage. ....	14
Tableau II.1 : Tableau de vérité des différents opérateurs logique. ....	27
Tableau III.1 : Tableau descriptif des faits conçus dans notre système. ....	53
Tableau III.2 : Identification des cas d'utilisation. ....	57
Tableau III.3 : Récapitulatif des scénarios par tâches de l'administrateur. ....	61
Tableau III.4 : Récapitulatif des scénarios par tâches de l'expert. ....	62
Tableau III.5 : Récapitulatif des scénarios par tâches du propriétaire. ....	63
Tableau A.1: Les différentes multiplicités d'une association. ....	101

# *Introduction Générale*

---

## *Introduction générale*

La région nord algérienne est connue pour une importante activité sismique qui est due principalement à la nature géologique de la région maghrébine et à ses caractéristiques tectoniques à la frontière des plaques africaine et eurasienne, en mouvement compressif permanent. Les images de désordre après chaque tremblement de terre majeur nous incitent à prendre conscience des dommages qu'il engendre dans les constructions et surtout les vies humaines.

Lors de la survenue d'un séisme, les autorités en charge de la gestion de crises doivent pouvoir dresser aussi rapidement que possible un « paysage » de la situation dégradée, afin d'entreprendre des actions répondant aux besoins immédiats.

À ce jour, en Algérie comme dans la plus part des autres pays touchés par ce type de catastrophe, la gestion de crise est menée par des méthodes dites classiques fondées sur un guide technique, l'expérience acquise après les différents séismes « El-Asnam, 1980), (Constantine, 1985), (Tipaza, 1989), (Ain-Temouchent, 1999) et (Boumerdès, 2003) » montre que ce mode d'évaluation n'est pas approprié dans ce cas, pour divers raisons, parce que :

- C'est un sondage minutieux qui a besoin de beaucoup de temps.
- Le mode d'évaluation demande un nombre important d'enquêteurs « ingénieurs, architectes, formateurs, experts, ... ».
- une bonne évaluation exige une expérience dans le domaine d'expertise des dommages.

À cet effet, l'expérience affirme que les gestionnaires ont besoin de disposer d'outils capable d'identifier rapidement l'ampleur des conséquences occasionnées.

Le problème rencontré dans la phase post-sismique ne se limite pas qu'à la maîtrise des enquêteurs et au temps d'inspection, l'expérience endurée nous a enseigné aussi que l'une des causes les plus meurtrières est liée à la nécessité d'utiliser le bâtiment pendant la situation d'urgence sismique par les occupants. Cependant, l'absence de la société dans l'organisation post-sismique rend la situation propice à l'effondrement, et la compétence de la population pour une bonne contribution à la gestion de crise nécessite l'utilisation d'outils performants qui représentent la réalité de leurs constructions.

Nous nous intéressons dans ce travail à la réalisation d'un outil logiciel qui assistera d'une part, les ingénieurs du génie civil confrontés aux problèmes d'évaluation des dommages subis par les constructions en cas d'un tremblement de terre, et d'autre part, les occupants des constructions à savoir le niveau de dommage global subis par celles-ci.

Devant un tel problème, il était question de faire un choix judicieux sur les moyens à utiliser pour aboutir à des résultats satisfaisants.

Après une analyse de ce problème, il s'est avéré que la démarche d'aide à l'expertise des constructions fait appel à des heuristiques et à l'expérience d'un expert en génie parasismique. En effet l'établissement d'un diagnostic tient plus de savoir de l'expert, que d'une démarche stricte et univoque, dans la mesure où il utilise ses compétences

professionnelles pour pouvoir estimer le degré des dommages d'une construction et par conséquent décider sur son sort (réparation ou démolition).

De ce fait, une approche procédurale est difficile à envisager par contre une approche système expert (technique de l'intelligence artificielle) s'avère mieux adaptée car un système expert est un programme qui tente de reproduire les facultés de décision ou de jugements de valeur des experts humains dans un domaine restreint.

Pour mieux faciliter la présentation de notre travail, nous l'avons réparti en quatre chapitres :

- Le premier chapitre est dédié à la définition et la description du domaine d'expertise ;
- Le second chapitre introduit les généralités sur les systèmes experts ;
- Le troisième chapitre est consacré à l'analyse et la conception de notre système ;
- Le dernier chapitre décrit les moyens et techniques utilisés pour sa réalisation, ainsi que quelques tests.

*CHAPITRE* I

---

***GÉNÉRALITÉS  
SUR LE  
DOMAINE  
D'EXPERTISE***

**Introduction :**

Le génie civil est un domaine spécialisé et très vaste. Il s'agit du domaine de l'urbanisme dont la fonction est de construire et, d'une manière plus générale, de mettre en œuvre les décisions d'aménagement du territoire. Et comme tout autre domaine, il est discrétisé (fragmenté) en plusieurs disciplines, principalement on retrouve : la structure, l'hydraulique, l'environnement et la géotechnique.

On distingue au sein du génie civil : le bâtiment et les travaux publics. Ces domaines ne sont pas antonymes, mais les besoins de l'économie ont fait que ces spécialités se sont affûtées au fil du temps. Quelle que soit sa spécialisation, l'ensemble du travail d'un ingénieur civil repose sur deux concepts clés : l'évaluation des charges et la résistance des matériaux. En fait, l'objectif premier est de combiner un ensemble de matériaux afin de résister à des charges pour accomplir une fonction principale de protéger les occupants des différentes agressions en particulier le séisme.

Le cadre de notre travail se limite à l'étude des constructions de type bâtiments, plus précisément on s'est intéressé à l'évaluation de leurs dommages après tremblement de terre.

**I. Généralités sur le domaine d'application : [18]****I.1. Éléments structuraux :****I.1.A. Élément principale:**

Faisant partie du système de contreventement (poutre, poteau, plancher, voile, mur...).

**I.1.B. Élément secondaire:**

Élément porteur ne faisant pas partie du système de contreventement (poteaux, murs...).

**I.2. Éléments non structuraux :**

Éléments n'ayant pas de fonction porteuse ou de contreventement (cloisons, acrotères, auvents...).

**I.3. Système de contreventement :**

Ensemble d'éléments de construction assurant la rigidité et la stabilité vis-à-vis des forces horizontales engendrées par le séisme.

**I.4. Mur de contreventement :**

Élément identique au voile de contreventement mais réalisé en maçonnerie chaînée.

**I.5. Voile de contreventement :**

Élément porteur rigide en béton armé destiné à transmettre les charges latérales dans les fondations.

**I.6. Système porteur :**

L'ensemble des éléments verticaux qui reprennent uniquement les charges verticales (poids propres de la construction, les charges d'exploitations, ...).

**I.7. Construction en Portiques :**

Structure composée de poteaux et de poutres rigidement liés.

**I.8. Construction en maçonnerie :**

Structure composée uniquement de murs en maçonnerie.

**I.9. Règlement parasismique algérien (RPA) :**

C'est un ensemble des normes et règles parasismiques élaborées par les pouvoirs publics suite au séisme d'El Asnam 1980. Il a été mis en application en 1983 et a fait objet de plusieurs modifications, dont la plus récente remonte à 2003 suite au séisme de Boumerdès, ces normes et règles doivent être prise en considération par les ingénieurs pour réaliser des constructions parasismiques.

**II. Bâtiments :****II.1. Définition :**

Les bâtiments sont destinés à servir d'abri, c'est-à-dire à protéger des intempéries des personnes, des biens et des activités.



**Figure 1: Ensemble des bâtiments.**

**II.2. Typologies des bâtiments :**

Dans tous les règlements, la typologie d'une construction type bâtiment est désignée par son système de contreventement, à cet effet on distingue :

- Bâtiments en maçonnerie ;
- Bâtiments en poteaux-poutres (portique) ;
- Bâtiments en voiles ;
- Bâtiments en charpente métallique ;
- Bâtiments mixtes.

**Remarque :** Seuls les bâtiments en maçonnerie et en portique sont considérés dans cette étude.

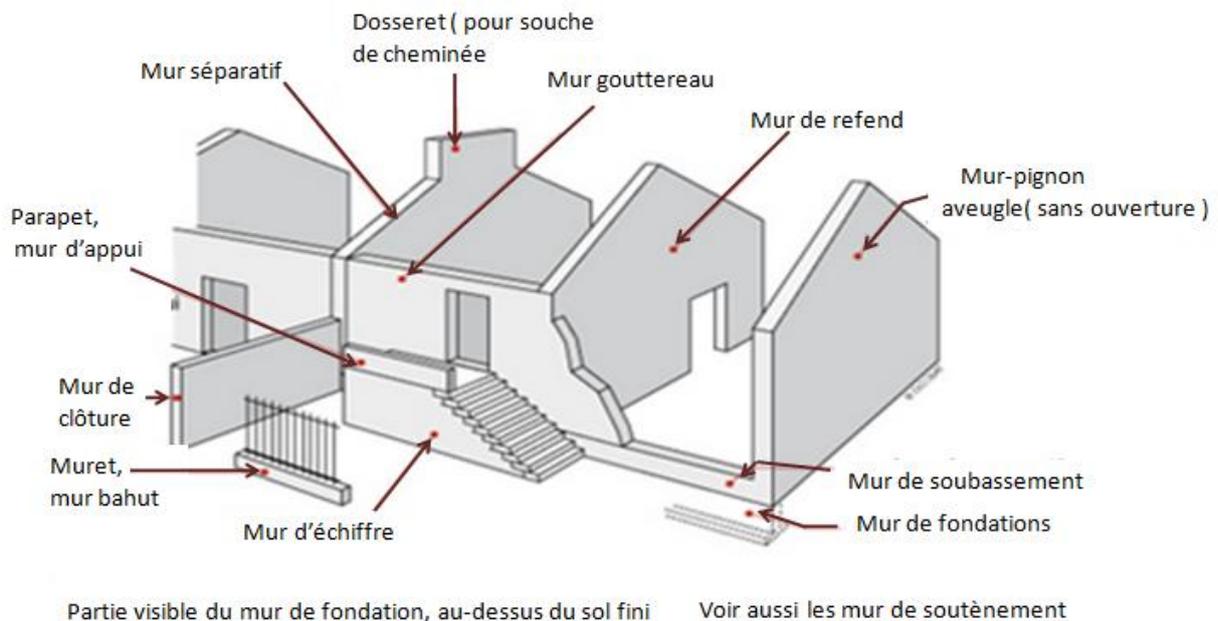
**II.3. Composants des bâtiments :**

Sommairement la composition des bâtiments est la même, on distingue deux types de composants : éléments structuraux (principaux) et d'autres d'éléments non structuraux (secondaires).

**II.3.A. Bâtiments en maçonnerie :**

Éléments principaux		Éléments secondaires	
Élément porteur	mur en maçonnerie	Cage d'escalier	
Élément de contreventement	mur en maçonnerie	Éléments de remplissage	Murs en maçonnerie.
Éléments Planchers	Dalle en corps creux, dalle	Éléments intérieurs	Cloisons, plafond, élément vitre.
Élément Toiture	terras (dalle), toiture incliné. (bois-tuiles).	Éléments extérieur	Balcon, auvent, acrotère, cheminé, garde corps.

**Tableau I.1 : Éléments d'un bâtiment en maçonnerie.**



**Figure 2 :Représentation des composants du bâtiment en maçonnerie.**

## II.3.B. Bâtiments en portique :

Éléments principaux		Éléments secondaires	
Élément porteur	Poteaux en Béton armé	Cage d'escalier	
Élément de contreventement	Portiques en Béton armé	Éléments de remplissage	Mur en maçonnerie, mur préfabriqué,
Éléments Planchers	Dalle en corps creux	Éléments intérieurs	Cloison, élément vitre, plafond.
Élément Toiture	Terrasse (dalle), toiture inclinée. (bois-tuiles).	Éléments extérieur	Balcon, auvent, acrotère, cheminé, garde corps.

Tableau I.2 : Éléments d'un bâtiment en portique.

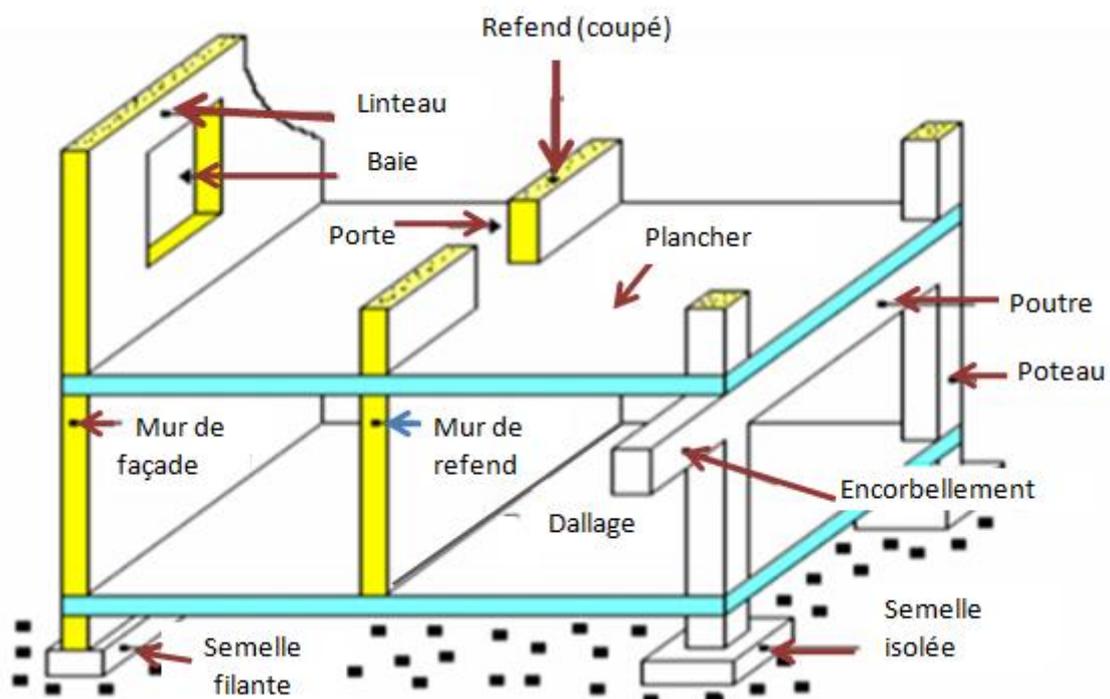


Figure 3 : Représentation des composants d'un bâtiment en poteaux-poutres.

### III. Tremblement de terre (séisme):

#### III.1. Définition :

Le séisme est un phénomène lié à l'existence de mouvements tectoniques continuellement en action qui soulèvent des chaînes de montagnes et creusent des tranchées océaniques à la surface de la terre.

Ces mouvements affectent une quinzaine de plaques et engendrent des contraintes sur les lignes de contact des plaques. Lorsque ces contraintes deviennent trop élevées et supérieures à la résistance au frottement, une rupture brutale se produit : c'est un tremblement de terre, à la suite duquel un nouvel état provisoire de stabilité est réalisé.

Il résulte de cette description qu'il existe des régions du monde à plus ou moins haut risque sismique, suivant leur localisation géographique par rapport à ces zones de jonctions des plaques tectoniques.

#### III.2. Paramètres d'un séisme : [18]

##### III.2.A. Foyer d'un séisme :

C'est le point origine de la rupture au sein de l'écorce terrestre engendrant le séisme.

##### III.2.B. Epicentre :

L'épicentre est un point situé à la surface du sol à la verticale du foyer.

##### III.2.C. Intensité :

L'intensité mesure l'importance d'un séisme en un lieu donné d'après les manifestations ressenties par la population et les dégâts qu'il provoque.

##### III.2.D. Accélération d'un séisme :

Fait correspondre à chaque intensité du séisme ses effets macrosismiques, autrement dit les dommages engendrés par les différentes classes de constructions.

##### III.2.E. Faille :

Une faille est une cassure de l'écorce terrestre qui partage un ensemble rocheux en deux compartiments décalés.

##### III.2.F. Ondes sismiques :

Elles correspondent aux ondes élastiques qui se propagent à la surface et à l'intérieur de notre planète à la suite d'un séisme.

##### III.2.G. La magnitude :

C'est une grandeur mesurant l'énergie libérée lors d'un séisme.

##### III.2.H. Classification des séismes :

Les séismes sont classés suivant leurs intensités et leurs accélérations :

- **Séisme faible** : Intensité modérée mais significative, il correspond aux intensités comprises entre [0, 3] sur l'échelle de Richter et à une accélération de l'ordre de (0.15g).
- **Séisme modéré** : Intensité élevée, il correspond aux intensités comprises entre [3, 5] sur l'échelle de Richter et à une accélération de l'ordre de (0.25g).

- **Séisme fort** : Intensité très élevée, il correspond aux intensités comprises entre [5, 7] sur l'échelle de Richter et a une accélération de l'ordre de (0.50g) ;

La figure suivante illustre quelques paramètres d'un séisme :

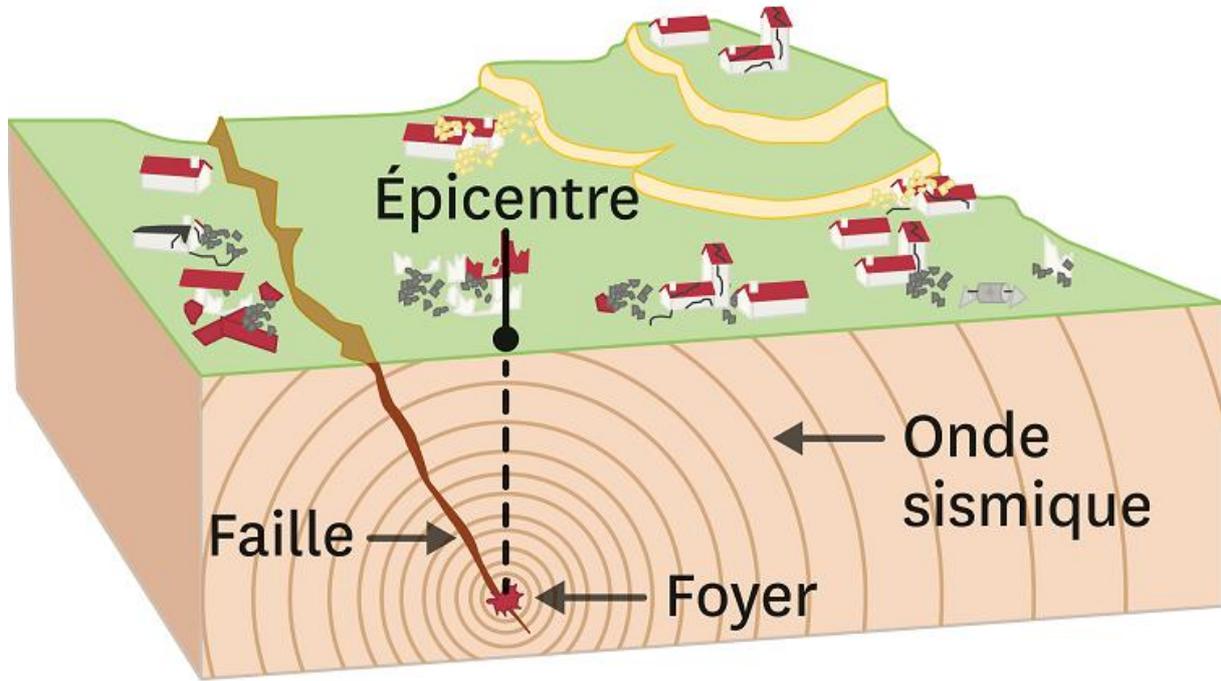


Figure 4 : Schéma illustrant les paramètres d'un séisme.[21]

### III.3. Les séismes en Algérie :

L'Algérie est au centre de deux plaques tectoniques séparées par la faille sud-atlasique. Au nord, se trouve la tectonique alpine et au sud, la plate-forme saharienne, qui est assez stable.

Une centaine de séismes de faible et de moyenne intensité se produisent, chaque mois, en Algérie. En fait, nos régions côtières enregistrent une importante activité sismique et chaque séisme ressenti fait rappeler aux populations les tremblements de terre dévastateurs. Parmi les séismes violents qu'on peut citer :

Date	Epicentre	Magnitude	Morts et blessés
mai 1716	Wilaya d'Alger	/	20 000 morts, des milliers de blessés
2 mars 1825	Wilaya de Blida	/	7 000 morts, des milliers de blessés

16 novembre 1869	Wilaya de Biskra	/	40 morts, des dizaines de blessés
6 août 1947	Wilaya d'Annaba	5,3	3 morts, des centaines de blessés
9 septembre 1954	Wilaya de Chlef	6,8	1 250 morts, 3 000 blessés
1 <sup>er</sup> janvier 1965	Wilaya de M'Sila	5,5	4 morts, 350 blessés
10 octobre 1980	Wilaya de Chlef	7,7	5 000 morts, 9 000 blessés
27 octobre 1985	Wilaya d'Annaba	5,9	6 morts, des dizaines de blessés
29 octobre 1989	Wilaya de Tipaza	5,9	30 morts, 245 blessés
21 mai 2003	Wilaya de Boumerdès	6,8	2 266 morts, 10 261 blessés
27 mai 2003	Wilaya d'Alger	5,8	9 morts, 180 blessés
14 mai 2010	Wilaya de Bouira	5,1	2 morts, 43 blessés
1 <sup>er</sup> août 2014	Wilaya d'Alger	5,6	
29 mai 2016	Wilaya de Médéa	5,3	
5 mars 2017	Wilaya de Skikda	4,6	

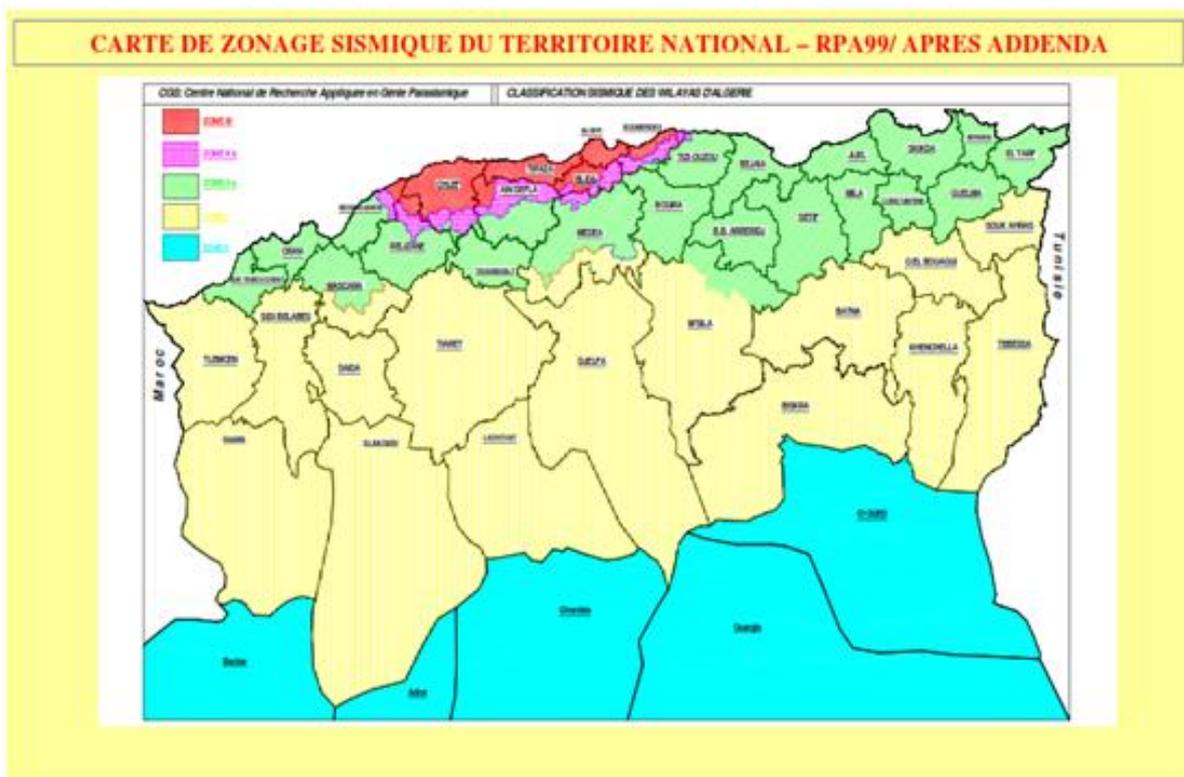
**Tableau I.3 : Historique des séismes en Algérie.**

**III.4. Zonage sismique : [18]**

Le zonage sismique correspond à affecter à chaque commune de l'Algérie une sismicité probable. Etabli par le CGS (Centre national de recherches appliquées en Génie paraSismique), suivant la classification établie dans le RPA 99 version 2003 on trouve les quatre zones suivantes :

- Zone 0 : sismicité négligeable.
- Zone I : sismicité faible.
- Zone II<sub>a</sub> et II<sub>b</sub> : sismicité moyenne.
- Zone III : sismicité élevée.

La figure suivante illustre le zonage sismique établi par le RPA pour le territoire national :



**Figure 5 : Zonage sismique du territoire national établi par le RPA 99 V2003.[18]**

Les différentes zones sismiques sont représentées dans la carte ci-dessus par les couleurs suivantes :

- Couleur bleu : Zone 0 ;
- Couleur jaune : Zone I ;
- Couleur verte : Zone II<sub>a</sub> ;
- Couleur rose : Zone II<sub>b</sub> ;
- Couleur rouge : Zone III.

**IV. Moyens utilisés pour l'évaluation des dommages post sismique :**

Un formulaire a été mis en œuvre peu de temps après le séisme dévastateur d'El Asnam en 1980, dans un contexte d'urgence sismique (El Asnam, 1980,...., Boumerdès, 2003), jusqu'au jour d'aujourd'hui, pour l'inspection des dommages causés par le séisme, ce formulaire représente l'outil indispensable des autorités (des services concernés : **CTC**: Centre de Contrôle Technique, **CGS**: Centre National de recherches appliquées en Génie parasismique, **CNERIB**: Centre National d'Etudes et de Recherches Intégrées du Bâtiment).

Son efficacité est représentée par : L'homogénéisation de l'inspection, Il offre une surveillance permanente de l'inspection.

Wilaya de Boumerdès							
Fiches d'évaluation des dommages							
séisme du 21 mai 2001							
code inspecteur :							
date :							
identification de la construction :							
Secteur	zone			Construction calculée au séisme :		oui	non
Adresse ou éléments d'identification				Construction contrôlée :		oui	non
Usage de la construction(*)							
Logement	scolaire		commerciale				
Administratif	hospitalier		industriel				
Socioculturel	sportif		réservoir d'eau				
Autre (à préciser)							
Description sommaire							
Age approximatif				vide sanitaire :		oui	non
Nombre de niveaux				Sous-sol :		oui	non
Nombre de joints de dilatation en elevation infrastructure				élément extérieur indépendant (escalier, auvent, passage, couvert)			
Problème de sol autour de la construction							
Faïlle :	oui	non	affaissement	<b>S</b>	soulèvement	oui	non
Liquéfaction :	oui	non	glissement			oui	non
Fondation-infrastructure							
Fondation							
Type de fondation :				infrastructure (dans le cas VS ou S/SOL)			
Type de dommage : <b>S</b>				voile béton continu <b>I</b> 1 2 3 4 5			
Tassement uniforme : oui non				poteaux béton avec remplissage 1 2 3 4 5			
Glissement : oui non							
Basculement : oui non							
Structure résistante							
Elément porteurs (charges verticales)				Elément de contreventement			
Murs en maçonnerie 1 2 3 4 5				Murs en maçonnerie 1 2 3 4 5			
Voile béton <b>V</b> 1 2 3 4 5				Voile béton 1 2 3 4 5			
Poteau métallique 1 2 3 4 5				Portiques béton armé <b>H</b> 1 2 3 4 5			
Poteau bois 1 2 3 4 5				Portique métalliques 1 2 3 4 5			
Autre 1 2 3 4 5				Palées triangulées 1 2 3 4 5			
				Autre 1 2 3 4 5			
Plancher-toiture terrasse							
Béton armé 1 2 3 4 5				Toiture inclinée			
Solives métallique 1 2 3 4 5				Charpente métallique 1 2 3 4 5			
Solive bois 1 2 3 4 5				Charpente bois 1 2 3 4 5			
				Couverture tuile 1 2 3 4 5			
				Couverture amiante ciment 1 2 3 4 5			
				Couverture métallique 1 2 3 4 5			
<b>R</b>							

Figure 6 : Le modèle du formulaire d'évaluation utilisé par les autorités algériennes (Partie 1).

Elément secondaires							
<u>Escaliers</u>							
Béton	1 2 3 4 5						
Métal <b>E<sub>ES</sub></b>	1 2 3 4 5						
Bois	1 2 3 4 5						
<u>Remplissages extérieurs</u>							
Maçonnerie	1 2 3 4 5						
Béton préfabriqué <b>E<sub>RE</sub></b>	1 2 3 4 5						
Bardages	1 2 3 4 5						
Autres	1 2 3 4 5						
<u>Autres éléments intérieurs</u>							
Plafonds	1 2 3 4 5						
Cloisons <b>E<sub>IN</sub></b>	1 2 3 4 5						
Elément vitrés	1 2 3 4 5						
<u>Eléments extérieurs</u>							
Balcons	1 2 3 4 5						
Garde cops <b>E<sub>EX</sub></b>	1 2 3 4 5						
Au vent	1 2 3 4 5						
Acrotère-corniche	1 2 3 4 5						
Autres	1 2 3 4 5						
Influence des constructions adjacentes							
La construction menace une autre construction	oui	non					
La construction est menacée par une autre construction	oui	non					
La construction peut être soutien par une autre construction	oui	non					
La construction peut être soutenu par une autre construction	oui	non					
Victimes							
Oui – non – peut être	si oui combien ??						
Commentaire sur la nature et la cause probable des dommages							
	<u>Sens transversal</u>			<u>Sens longitudinal</u>			
Symétrie en plan	bon	moyen	mauvais	<b>F</b>	bon	moyen	mauvais
Régularité en élévation	bon	moyen	mauvais		bon	moyen	mauvais
Redondances des files	bon	moyen	mauvais		bon	moyen	mauvais
Autres commentaires							
Evaluation finale							
<u>Niveau général des dommages</u>				<u>Couleur à utiliser</u>			
1 - 2 - 3 - 4 - 5				vert – orange - rouge			
<u>Mesures immédiates à prendre :</u>							

**Figure 7 : Le modèle du formulaire d'évaluation utilisé par les autorités algériennes (Partie 2).**

**V.Évaluation des dommages Globaux :**

Selon les dommages aux composants, l'inspecteur (l'expert) doit dériver (donner) la catégorie de dommages globaux de la structure. Selon ses dégâts globaux, chaque structure est ensuite marquée avec un code de couleur spécifique, de sorte que:

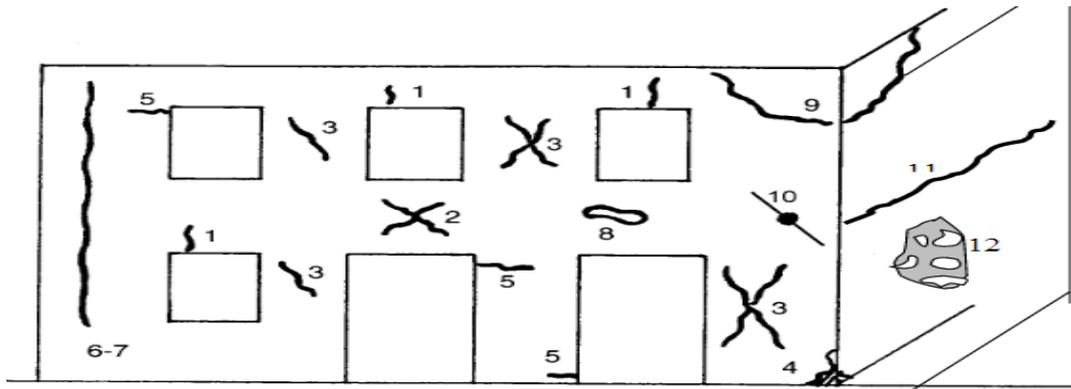
- Le bâtiment peut toujours rester en service si le dégât global correspond à une légère catégorie de dégâts 1 ou 2, mentionnés par des étiquettes vertes "claires" ou "foncées". Cependant, il peut être nécessaire de procéder à de légères réparations ultérieurement ;
- Le bâtiment doit être évacué si les dégâts globaux correspondent à des catégories de dommages importants 3 ou 4, mentionnés par des étiquettes orange "claires" ou "foncées". Il faudra un renforcement et une réparation importants avant de retourner en service ;
- La structure doit être démolie pour une catégorie de dommages globaux 5, c'est-à-dire mentionnés par la balise "Rouge".

Le tableau suivant nous illustre les différents degrés de dommages qui peuvent être subis par une construction, cette classification a été établie par l'organisme de CTC (contrôle technique des constructions en Algérie) :

<b>Dommage</b>	Aucun	Léger	Modéré	Important	Destruction (totale/partielle)
Pourcentage	0%	25%	50%	75%	100%
Valeur	1	2	3	4	5
Couleur	Vert clair	Vert foncé	Orange clair	Orange foncé	Rouge
Symbole	V1	V2	O3	O4	R5

**Tableau I.4 : Échelle d'évaluation de degré de dommage.**

### V.1. Orientation sur l'évaluation des dommages causés par le séisme :



**Figure 8 : Schéma de référence pour les fissures dans la maçonnerie.**

Les différents numéros dans la figure représentent les fissures correspondantes suivantes :

- 1: fissures presque verticales sur les linteaux d'ouverture;
- 2: fissures diagonales dans les poutres (parapets de fenêtre, linteaux);
- 3: fissures diagonales dans les éléments verticaux (piliers de maçonnerie);
- 4: concassage local de la maçonnerie avec ou sans évacuation des matériaux;
- 5: fissures presque horizontales au sommet et / ou au pied des piliers de maçonnerie;
- 6: fissures presque verticales aux intersections des murs;
- 7: identique à 6 mais avec fissure traversant;
- 8: évacuation des matériaux au niveau des appuis de poutre en raison d'un battage;
- 9: formation d'un coin déplacé à l'intersection de deux parois orthogonales;
- 10: défaillance des tirants ou glissement de la liaison;
- 11: fissures horizontales au niveau du sol ou au niveau du grenier;
- 12: séparation de l'un des éléments d'un mur double.

### V.2. Procédure d'inspection des dommages causés par les séismes :

#### V.2.A. Inspection détaillée des dommages sismiques :

- **D1 légers dégâts :**

C'est un dommage qui n'affecte pas de manière significative la capacité de la structure et ne met pas en péril la sécurité des occupants, due à la chute des éléments non structurels. Le dommage est léger lorsque la chute des objets peut être immédiatement évitée.

- **Maçonnerie :**

Des fissures de largeur  $\leq 1$  mm, indépendamment de leur répartition dans les murs de maçonnerie et dans les sols, sans éloignement des matériaux,

séparation limitée ou légères dislocations ( $\leq 1$  mm) entre parties de structures, par exemple entre murs et planchers ou entre murs et escaliers ou entre Orthogonaux. Limitée à l'aplomb, non associée à des phénomènes induits par le séisme de séparation des murs en élévation ou en établissements, qui, s'ils existent, doivent être préexistants et ne pas influencer la capacité structurelle. Dommages limités sur les toits les plus flexibles (bois ou acier) avec la chute conséquente de certaines tuiles sur les bords. Chute de petites portions de plâtre dégradé ou de stuc, non relié à la maçonnerie.

- **Béton armé :**

Légères fissures dans les poutres (jusqu'à 1 mm), fissures répandues mais non verticales ( $<0,5$  mm) en colonnes ou en cloisons. Fissures jusqu'à 2 mm en raison de la séparation des parois de remplissage des structures, légères fissures diagonales dans les remplissages ( $<1$  mm).

- **D2 - D3 dégâts moyens-graves**

C'est un dommage qui modifie sensiblement la capacité de la structure, sans se rapprocher de la limite d'effondrement partiel des principaux composants structurels. Possibilité de chute d'objets non structurels.

- **Maçonnerie :**

Des fissures plus sévères par rapport à D1, également avec l'expulsion de matériel et quelques millimètres de largeur (jusqu'à environ 1 cm) ou plus large, si près des ouvertures, des symptômes de fissures dues au concassage, des séparations importantes entre planchers et / ou escaliers et murs Et entre les parois orthogonales, quelques effondrements partiels dans les poutres secondaires des planchers.

Fissures de quelques mm dans les voûtes, et / ou avec des symptômes d'écrasement.

Dans les toits en bois ou en acier avec la couverture de tuiles, les dommages dans les poutres secondaires et la chute d'une partie significative de la couverture de tuiles. Visible hors plomb, induit par le tremblement de terre, mais en tout cas pas plus d'environ 1%.

- **Béton armé :**

Les fissures de flexion dans les poutres jusqu'à 4-5 mm, les fissures dans les colonnes et dans les murs de cisaillement jusqu'à 2-3 mm, le début du flambage des barres comprimées dans les colonnes, avec l'écaillage de la couverture en béton, juste résiduel perceptible.

Fissures évidentes ( $> 2$  mm) dans les murs de remplissage en raison de la séparation de la structure, des fissures diagonales jusqu'à quelques mm, un

écrasement évident aux coins en contact avec les structures de support, parfois avec une expulsion localisée de matériel.

- **D4 – D5 dégâts très lourds :**

C'est un dommage qui modifie significativement la capacité de la structure, la rapprochant de la limite d'effondrement partiel ou total des principaux composants structurels. Cet état se caractérise par des dommages plus lourds que les précédents, y compris l'effondrement.

La figure ci-dessus illustre les différentes catégories de dommages citées avant :

Echelle EMS98	1	2	3	4	5
Structure en maçonnerie					
Structure en béton armé					

**Figure 9 : Catégorie de dommage pour les constructions en maçonnerie et en Béton Armé.**

### V.2.B. Inspection suivant la fiche d'évaluation du CTC ou CGS :

Les dommages maximaux qui affectent les composants structurels: fondations, poutres, colonnes et armoires, toits et dalles, nommés ici comme «groupe structurel» désigné «I», «V», «H» et «R». Pour chacun de ces quatre composants "structurels" correspond son endommagement spécifique dans l'ensemble {1; 2; 3; 4; 5}, en fonction de ses dégâts observés.

Les dommages maximaux qui affectent les composants secondaires: les escaliers, les plombages extérieurs, les panneaux intérieurs, les éléments extérieurs tels que les cheminées, les parapets et les balcons, nommés ici «groupe non structurel» désigné «E<sub>SS</sub>», «E<sub>RE</sub>», «E<sub>IN</sub>» et «E<sub>EX</sub>». A chacun de ces quatre composants "secondaires", c'est-à-dire "non structurel", correspond à son dommage spécifique dans l'ensemble {1; 2; 3; 4; 5}, en fonction de ses dégâts observés.

Le désordre géotechnique (tels que les failles, la liquéfaction, les glissements de terrain ...) nommés ici «groupe qualitatif» est désigné «S». Cependant, pour la présente étude, ce paramètre n'est pas considéré comme un paramètre de régulation.

**VI. Conclusion :**

Nous avons présenté dans ce chapitre le domaine d'expertise qu'est l'étude des dommages post-sismiques.

Après quelques définitions concernant ce domaine, nous avons présenté un moyen permettant d'évaluer les dommages post-sismiques qu'est la fiche technique d'évaluation utilisée par le CTC et le CGS depuis le séisme d'EL ASNAM.

Cette fiche établit un véritable diagnostic d'une construction qui aboutit à une évaluation générale de l'état de celle-ci.

Nous aborderons dans le prochain chapitre « généralités sur les systèmes experts », qui est la base de notre projet.

*CHAPITRE* **II**

---

***GÉNÉRALITÉS  
SUR LES  
SYSTÈMES  
EXPERTS***

**Introduction :**

Les experts humains sont capables d'effectuer un niveau élevé de raisonnement grâce à leur grande expérience et connaissance sur leurs domaines d'expertise. Les systèmes expert sont apparus dans les années 70, ce sont des programmes conçus pour simuler le savoir faire d'un expert dans un domaine précis, pour cela ils utilisent la connaissance correspondante au domaine afin de fournir une performance comparable à l'expert humain. En général, les concepteurs de systèmes experts effectuent l'acquisition des connaissances grâce à un ou plusieurs interviews avec l'expert du domaine. Aujourd'hui, les systèmes experts constituent une technologie bien définie faisant partie des systèmes à base de connaissances.

Le but de ce chapitre est d'introduire les concepts de base des systèmes experts, leurs caractéristiques, les différents formalismes de représentation des connaissances, leurs fonctionnements ainsi que leurs mécanisme d'inférence.

**I. Généralités sur les systèmes experts :****I.1. Définition d'un système expert : [11]****I.1.A. Définition 1 :**

Un système expert est un système logiciel informatique particulier, qui est capable de donner sur un sujet particulier un conseil égal en qualité à celui qui pourrait être donné par un expert humain reconnu.

**I.1.B. Définition 2 :**

Les systèmes experts sont des programmes conçus pour raisonner habilement à propos des tâches dont on pense qu'elles requièrent une expertise humaine considérable.

**I.1.C. Définition 3 :**

Les systèmes experts sont des programmes résolvant des problèmes qui sont habituellement résolus par des experts humains. Pour cela, ils requièrent un accès à une base de connaissances conséquente qui doit être construite de façon efficiente. Il doivent être à même de fournir divers modes de raisonnement et de justifier les conclusions auxquelles ils aboutissent.

Il ressort des définitions précédentes qu'un système expert :

- est un logiciel,
- qui contient une base de connaissances sur un sujet particulier, et
- qu'il est capable de fournir sur ce sujet un conseil égal en qualité à celui que pourrait donner un expert humain reconnu.

**I.2. Historique des systèmes experts : [14]**

Le premier système expert fut Dendral en 1965, créé par les informaticiens Edward Feigenbaum, Bruce Buchanan, le médecin Joshua Lederberg et le chimiste Carl Djerassi.

Il permettait d'identifier les constituants chimiques d'un matériau à partir de spectrométrie de masse et de résonance magnétique nucléaire, mais ses règles étaient

mélangées au moteur. Il fut par la suite modifié pour en extraire le moteur de système expert nommé Meta-Dendral.

Le plus connu, peut-être, fut Mycin en 1972-73, système expert de diagnostic de maladies du sang et de prescription de médicaments, avec un vrai moteur et une vraie base de règles. Cependant ses règles étaient affectées de coefficients de vraisemblance qui donnaient à chacune d'entre elles un poids particulier face aux autres. Le moteur produisait un chaînage avant simple tout en calculant les probabilités de chaque déduction, ce qui le rendait incapable d'expliquer la logique de son fonctionnement et de détecter les contradictions. Quant aux experts, ils étaient obligés de trouver des coefficients de vraisemblance pour chacune des conclusions de leurs règles, une démarche compliquée et antinaturelle qui déniait leur capacité de raisonnement.

Les systèmes experts ont eu leur heure de gloire dans les années 1980, où on a trop rapidement pensé qu'ils pourraient se développer massivement. En pratique, le développement de ce genre d'application est très lourd car, lorsque l'on dépasse la centaine de règles, il devient difficile de comprendre comment le système expert « raisonne » (manipule faits et règles en temps réel), et donc d'en assurer la mise au point finale puis la maintenance.

Le projet Sachem (Système d'Aide à la Conduite des Hauts fourneaux En Marche, chez Arcelor), opérationnel dans les années 1990, est l'un des derniers projets « système expert » à avoir vu le jour. Il est conçu pour piloter des hauts-fourneaux en analysant les données fournies en temps réel par un millier de capteurs. Le projet a coûté entre 1991 et 1998 environ 30 millions d'euros, et le système économise environ 1,7 euros par tonne de métal.

### **I.3. Caractéristiques des systèmes experts : [13]**

Les systèmes experts sont censés assurer le stockage de la connaissance, et raisonner sur cette dernière en utilisant des méthodes et des modes de résolution proches du raisonnement humain.

Les systèmes experts se caractérisent alors par les éléments suivants :

#### **I.3.A. Le langage de représentation de la connaissance :**

Un SE permet de représenter la connaissance nécessaire à la résolution de problèmes complexes relatifs à un domaine précis. Plusieurs formalismes de représentation de la connaissance existent dont les plus utilisés sont :

- Le formalisme à base de règles de productions ;
- Le formalisme à base d'objets.

#### **I.3.B. Posséder les facultés de :**

- **Raisonnement sur les connaissances** : Il s'agit d'appliquer des méthodes de raisonnement sur les connaissances existant en vue d'aboutir à des solutions.
- **Apprentissage** : c'est la capacité d'un SE à enrichir sa connaissance par celle déduite à l'issue de son activité de raisonnement.
- **Modularité** : L'ajout et/ou la modification des connaissances ne doit pas remettre en cause la stabilité du système.

- **Dialogue (système de communication)** : l'interaction du système avec son environnement (utilisateur et/ou développeur) doit être facilitée grâce à des interfaces conviviales.

 **Remarque :**

Les SE à base de règles de production sont les plus répandus. Nous expliquons leur architecture dans ce qui suit.

#### **I.4. Architecture des systèmes experts :**

Les systèmes experts se composent principalement de :

- Une base de connaissance ;
- Un moteur d'inférence ;
- Des interfaces ;
- Un module d'acquisition de connaissances ;
- Un module d'explication.

##### **I.4.A. La base des connaissances (BC) : [5]**

La base de connaissances dispose de savoir (fait) et de savoir-faire (règles) de l'expert au sein d'un domaine particulier. Elle contient deux éléments :

- **La base de faits (BF) :**

La base de faits est la mémoire de travail du système expert, elle est variable au cours de l'exécution et vidée lorsque l'exécution se termine. On l'appelle aussi connaissance assertionnelle.

La base de faits intègre deux types de faits :

- les faits initiaux (relativement à un cycle de base avant tout début de fonctionnement du moteur d'inférences) du domaine.
- les faits déduits par le moteur d'inférences qui sont propres au cas traité.

**Exemple :** le ciel est bleu.

- **La base de règles (BR) :**

C'est une partie indépendante de connaissances déductives exprimée par les règles de productions. Elle représente le savoir-faire de l'expert dans un domaine précis. On l'appelle aussi connaissance opératoire.

Les règles sont habituellement représentées par :

**SI « condition » ALORS « conclusion »**

- o **condition** : est une prémisse qui peut être une conjonction de propositions logiques.
- o **Action** : elle s'agit de l'action à effectuer si la situation décrite dans la condition de la règle est vérifiée. Ces conclusions indiquent les nouveaux faits déduits.

**Exemple :** Si le fonctionnaire est âgé de 60 ans alors il peut sortir en retraite.

#### **I.4.B. Le moteur d'inférence (MI): [13]**

Le moteur d'inférences définit le mécanisme de raisonnement d'un système expert. C'est à lui qu'incombe la sélection des connaissances et l'application des règles de production en vue de la résolution d'un problème donné.

#### **I.4.C. Les interfaces : [5]**

Les interfaces concrétisent et facilitent le dialogue homme/machine. Un SE offre deux types d'interfaces :

- **L'interface « Expert » :**

C'est une interface réservée à l'expert afin de faciliter la mise à jour des règles et la vérification des connaissances, elle est liée directement à l'interface d'acquisition.

- **L'interface « Utilisateur » :**

C'est grâce à cette interface que les utilisateurs puissent consulter les systèmes experts en introduisant des données (faits) de leurs problèmes et attendre une réponse.

Elle est mise à la disposition de l'utilisateur afin de lui rendre transparent la façon dont le système est conçu.

#### **I.4.D. Le module d'acquisition des connaissances : [2]**

L'objectif de ce module est de capturer aisément les unités de savoir-faire et de donner à l'expert la possibilité de modifier fréquemment la base de connaissances qui est alors rendue accessible.

Le module d'acquisition de connaissances offre une interface d'acquisition des connaissances et un module de contrôle de cohérence.

- **L'interface d'acquisition :**

Offre au spécialiste du domaine (expert), le moyen de saisir et de structurer ses règles selon une syntaxe bien définie.

- **Le module de contrôle de cohérence :**

Ce mécanisme de contrôle de cohérence de la base de connaissances est conçu pour remédier aux problèmes de contradictions éventuelles entre les connaissances (faits).

Ainsi, l'intégrité des données stockées dans la base est garantie, et les problèmes de duplication (redondance) des connaissances sont exclus.

#### **I.4.E. Le module d'explication :**

Il joue un rôle d'assistance pour un utilisateur désirant avoir une explication détaillée sur la façon dont est obtenu le résultat renvoyé à l'issue d'un déclenchement du système, il permet aussi de restituer la trace du raisonnement suivi par le moteur d'inférence.

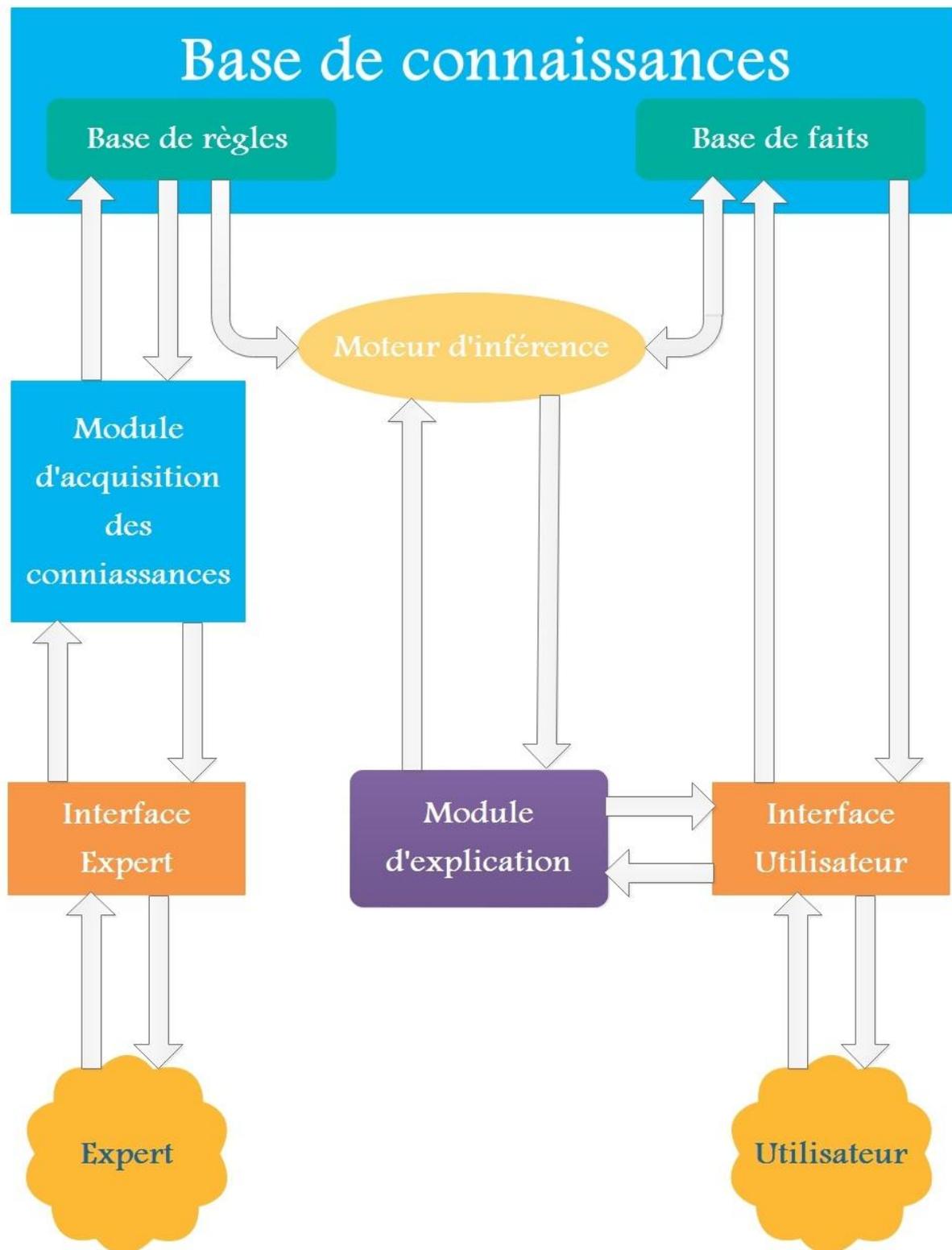


Figure 10: Architecture de base d'un système expert. [9]

**I.5. Principe des systèmes experts : [9]**

La principale particularité c'est que les connaissances du domaine sont isolés (faits et règles), ce qui fait du système expert un système basé sur la connaissance.

Ce ci s'appuie sur les connaissances d'un problème fournies au système qui fiabilise l'enchaînement approprié pour arriver à la solution, ainsi les connaissances sont déclarées d'une manière autonome et écrites indépendamment de leurs utilisations.

La connaissance est alors écrite indépendamment du raisonnement (son utilisation).



**Figure 11: Schéma simplifié illustrant le principe des systèmes experts. [5]**

**I.6. Typologie des systèmes experts : [1]**

Il existe deux types de systèmes experts :

**I.6.A. Systèmes expert dédié :**

C'est un système expert relatif à un domaine d'expertise particulier, il comprend une base de connaissance et un moteur d'inférence adéquat.

De tels systèmes peuvent être développés directement ou générés à partir d'un Shell.

**Exemple :**

- MYCIN : pour le diagnostic médical des maladies infectieuses du sang ;
- PROSPECTOR : pour la prospection minière et l'étude des possibilités d'existence de gisement de minerais.

**I.6.B. Générateurs de systèmes experts (Shell) :**

Un Shell est un environnement de développement de systèmes experts particuliers. Il comprend essentiellement un langage de représentation de connaissances et un moteur d'inférences.

Pour construire un système expert dédié à partir d'un Shell, revient à construire une base de connaissances d'un domaine particulier en utilisant le langage offert par ce dernier.

**Exemple :**

- OPS fondé sur le formalisme des règles de production ;
- VP-EXPERT : il est très répandu grâce à sa simplicité de maniement ;
- PAS-II : construit en vue d'évaluer dans quelle mesure les systèmes de production se prêtent à des formes d'apprentissage.

**I.7. Domaines d'application des systèmes experts : [13]**

Les domaines d'applications des SE sont principalement les suivants :

- **Analyse de situation** : expertise bancaire, financière, juridique, économique...

- **Diagnostic** : détection, diagnostic des pannes, diagnostic médical, pré-diagnostic, maintenance...
- **Aide à la décision** : identification, agrément de matériel ou de configuration de systèmes divers, classification...
- **Procédures** : aide à la mise en place et au suivi de procédures industrielles, administratives, comptables, organisationnelles, méthodologiques, etc....
- **Contrôle de Processus** : contrôle "intelligent" de processus industriels.
- **Formation** : Enseignement Assisté par Ordinateur.

En somme, un système expert peut être conçu dans tout domaine où une expertise humaine est prouvée.

### I.8. Objectifs des systèmes experts : [13]

Les systèmes experts sont sollicités comme réponse aux besoins d'informatisation des tâches intellectuelles permettant d'atteindre les objectifs suivants :

- Faciliter l'accès au savoir ;
- Capturer facilement les unités de savoir-faire des experts ;
- Partager le savoir-faire entre une poignée d'experts et un très grand nombre d'utilisateurs ;
- Utiliser la vitesse de calcul et le grand espace de stockage des machines actuelles pour simuler le comportement d'un expert humain ;
- Assister les humains dans le processus de décision ;
- Décortiquer le raisonnement expert pour expliquer.

### I.9. Avantages et inconvénients des systèmes experts : [11]

Les systèmes experts présentent un certain nombre d'avantages parmi lesquels on peut citer :

- Les systèmes experts sont évolutifs : la base de connaissances n'a pas besoin d'être complète avant de commencer à utiliser le système.
- L'enrichissement de l'expertise au sein de l'entreprise : lorsque la prise d'une décision nécessite l'intervention de plusieurs experts de domaines différents, un système expert pluridisciplinaire permet d'éviter de consulter tout le monde. Il en résulte un gain de temps considérable, et l'expertise de l'entreprise se trouve ainsi enrichie par la présence d'un tel outil.
- La transmission du savoir : le système expert sert de mémoire du savoir de l'entreprise. En effet il y a toujours risque de perte de compétences et de savoir faire (départs des experts, décès, ...), surtout lorsque la qualification exigée de l'expert est élevée. Les experts ayant déjà travaillé dans l'entreprise et ayant « transmis » leur savoir faire au système expert, la présence de celui-ci permet donc de mémoriser et de transmettre le savoir faire acquis par l'entreprise aux futurs cadres.

- La cohérence des décisions : un système expert permet la standardisation des décisions. La qualité du service et l'efficacité du personnel se verront ainsi améliorées.
- L'augmentation du volume d'affaires : les gains de temps résultant de l'informatisation du savoir faire de l'entreprise permettent une augmentation du volume d'affaires traitées, ceci se traduit généralement par l'amélioration de la rentabilité, ce qui constitue l'objectif clé de toute entreprise.

Malgré, les avantages qu'ils présentent, les systèmes experts possèdent toutefois certaines limites non négligeables :

- Ils ne peuvent pas remplacer les meilleurs spécialistes.
- La vue synthétique d'un expert ainsi que sa capacité de changer de point de vue en cours d'analyse ne sont pas prêts d'être simulées en machine.

## II. Représentation des connaissances :

La représentation des connaissances est une question essentielle en intelligence artificielle en général, et notamment dans les systèmes experts.

Le choix d'une bonne représentation est fondamental et détermine toute la suite de développement du système, de ce fait elle doit être aussi précise, aussi juste, aussi complète et aussi détaillée que celle possédée par un expert humain de domaine, et pour cela le choix d'un mode de représentation des connaissances est basé sur les critères suivants :

- **L'extensibilité** : le mode envisagé doit être souple, pour permettre une éventuelle modification du système ;
- **La simplicité et l'uniformité** : les concepts retenus ne doivent être complexes. Ils doivent permettre un accès aisé aux utilisateurs non avertis ;
- **La modularité** : chaque connaissance est un pas élémentaire de raisonnement et il n'y a pas d'effet direct d'une règle sur une autre. Ce qui garantit l'intégrité de la base en cas de mise à jour.
- **La rigueur** : le mode doit être un outil sûr pour la manipulation des connaissances, ce qui garantit la validité des résultats obtenus.

Représenter des connaissances consiste à trouver des structures d'informations qui réalisent une correspondance entre le monde réel et le système. De nombreuses recherches en intelligence artificielle ont conduit à l'élaboration de nombreux formalismes de représentation de la connaissance. On cite :

### II.1. Représentation logique :

Dans la logique classique on trouve :

#### II.1.A. Logique d'ordre 0 (logique propositionnelle) : [16] [3]

La logique d'ordre 0 traite des propositions qui sont en général des énoncés de faits. Une proposition est un énoncé auquel on peut associer une et une seule valeur de vérité. En général, la valeur de vérité sera, soit V(Vrai) soit F(Faux).

Les formules de cette logique sont formées selon certaines règles comme suit :

- **Base** : toute proposition est une formule ;
- **Induction** : si  $X$  et  $Y$  sont des formules, alors  $\neg X$ ,  $(X \wedge Y)$ ,  $(X \vee Y)$ ,  $(X \rightarrow Y)$ ,  $(X \leftrightarrow Y)$  sont des formules ;
- **Clôture** : une formule s'obtient uniquement à partir des règles de base et d'induction.

Le tableau ci-dessous présente les valeurs de vérités de différentes combinaisons entre connecteurs logiques :

<b>X</b>	<b>Y</b>	<b><math>\neg X</math></b>	<b><math>X \wedge Y</math></b>	<b><math>X \vee Y</math></b>	<b><math>X \rightarrow Y</math></b>	<b><math>X \leftrightarrow Y</math></b>
Vrai	Vrai	Faux	Vrai	Vrai	Vrai	Vrai
Vrai	Faux	Faux	Faux	Vrai	Faux	Faux
Faux	Vrai	Vrai	Faux	Vrai	Vrai	Faux
Faux	Faux	Vrai	Faux	Faux	Vrai	Vrai

**Tableau II.1 : Tableau de vérité des différents opérateurs logique.**

Dans la logique propositionnelle, on utilise les règles d'inférence (règles de dérivations) suivantes :

- **Modus Ponens** : si  $X$  et  $(X \rightarrow Y)$  alors on déduit  $Y$  ;
- **Modus Tonens** : si  $\neg Y$  et  $(X \rightarrow Y)$  alors on déduit  $\neg X$  ;
- **Enchaînement** : si  $(X \rightarrow Y)$  et  $(Y \rightarrow Z)$  alors on déduit  $(X \rightarrow Z)$ .

**Exemple :**

- Fait : Le soleil brille ;
- Règle : Si le soleil brille alors il fait beau.

### II.1.B. La logique d'ordre 1 (logique des prédicats) :

L'insuffisance du calcul propositionnel a donné naissance aux calculs des prédicats.

Le calcul des prédicats étend la logique propositionnelle grâce à l'introduction de variables et de quantificateurs universels ( $\forall$ ) lus « quelle que soit », et existentiel ( $\exists$ ) lus « il existe ».

On ajoute la règle suivante aux règles d'inférence citées précédemment :

- **La règle de spécification universelle** : Si  $p(X)$  est vraie pour tout  $X$  alors  $p(a)$  est vraie particulièrement pour une constante ( $a$ ).

- **Notion de prédicat** : c'est une fonction logique (booléenne) qui prend un ou plusieurs arguments, et retourne un résultat logique (soit vrai ou faux). Donc, il permet de vérifier l'existence d'une propriété pour un objet inconnu.

Cette propriété est soit :

- Vraie pour tout objet :  $\forall X P(X)$ .
- Vraie pour certains objets :  $\exists X P(X)$ .

**Exemple :**

- Fait : Homme(Socrate), i.e. : Socrate est un homme.
- Règle :  $\forall x (\text{Homme}(x) \rightarrow \text{Mortel}(x))$ , i.e. : tout homme est mortel.

- ✚ **Remarque** : la logique d'ordre 0 est inexploitable car elle ne permet d'exprimer toute la réalité, notamment quand on a besoin de préciser la portée de certaines assertions. Celle d'ordre 1 n'est plus utilisable à cause de l'explosion combinatoire rencontrée lors de la phase de filtrage du moteur d'inférence. En effet dans ce cas, le moteur d'inférence doit réaliser de l'unification générale i.e. la substitution de variables par d'autres mais non de semi unification i.e. la substitution de variables par des constantes (faits). C'est pour cela que dans la réalité on distingue :

### II.1.C. La logique d'ordre 1<sup>-</sup> :

Cela consiste à manipuler des variables (locales) dans les règles, par contre dans les faits on utilise la semi-unification qui désigne la substitution des variables par des constantes.

**Exemple :**

- Faits : ville(Alger) ;
- Règles : si femme(X) alors gentille(X).

### II.1.D. La logique d'ordre 0<sup>+</sup> :

Cette logique permet d'ajouter des items évaluables (comme variables globales) à la logique propositionnelle.

**Exemple :**

- Fait : La vitesse de la voiture  $\geq 100$  Km/h ;
- Règle : Si la température du malade est  $> 39^\circ\text{c}$  alors appeler le médecin.

### II.1.E. La logique de l'incertain :

Elle traite les informations incertaines en utilisant les règles de la logique d'ordre 0 et 0<sup>+</sup> entachée d'un degré d'incertitude.

**Exemple :**

- Si A alors (0.25) B. i.e. : (si A est certaine on peut associer un facteur de 0,25 de certitude à B)

**❖ Avantage de la représentation logique :**

Pour la logique du premier ordre, il a été prouvé qu'il existe des ensembles de règles (théorèmes et règles d'inférence) qui sont à la fois corrects et complets.

Ce résultat est la base de la puissance de la logique comme formalisme de représentation des connaissances. D'autre part, les arguments en faveur de la logique sont les suivants :

- La logique dispose de solides bases théoriques, pouvant s'étendre à de nouvelles utilisations du fait du développement actuel de logiques dites non-classiques telles que les logiques non-monotones, multivaluées, modales.
- Le caractère naturel avec lequel les éléments de connaissance peuvent être exprimés sous forme de formules logiques ; on peut exprimer un fait sans se soucier de ses manipulations.
- La flexibilité et la modularité dues à l'indépendance des assertions résultant du traitement déclaratif de la connaissance qui facilite la modification et l'élargissement d'une base logique de connaissances.

**❖ Inconvénients de la représentation logique :**

- Un des intérêts de la logique est que la cohérence et la complétude sont garanties mais cette caractéristique peut être, dans certains cas, un inconvénient car la logique est une représentation très formelle et mathématique. Elle ne permet pas, par exemple, une manipulation aisée des informations incertaines ou incomplètes.
- Aucune possibilité n'est offerte pour le contrôle, c'est à dire pour le choix des éléments de connaissance adéquats à employer dans une situation donnée. Le système doit le plus souvent travailler de façon exhaustive et manipuler un grand nombre d'hypothèses pour trouver celles utilisables. Contre la logique subsiste ainsi toujours l'argument de l'inefficacité due à l'explosion combinatoire. Si la base de connaissances devient trop importante, il n'est plus possible de l'utiliser de façon performante.

**II.2. Règles de productions : [11]**

La règle de production est la composante de représentation de connaissances d'un modèle général de raisonnement symbolique appelée système de production.

Une règle de production se présente sous la forme : « Si A alors B », elle peut traduire une relation, une information sémantique ou une action conditionnelle dans laquelle :

- A est la prémisse, c'est-à-dire la condition d'applicabilité de la règle, qui peut être une conjonction de propositions logiques de prédicats ou de relations, et

- B est la conséquence ou la conclusion, qui peut être une action à effectuer ou une autre proposition logique, laquelle peut à son tour, se trouver être prémisse d'une autre règle.

**Exemple :** S'il mange de la viande alors c'est un carnivore.

❖ **Avantages des règles de productions :**

La plupart des systèmes experts (par exemple : MYCIN) ont été développés en utilisant ce type de formalisme pour la représentation des connaissances, car il présente de nombreux avantages parmi lesquels :

- Sa facilité d'expression ;
- Sa large diffusion : les logiciels utilisant cette technique sont nombreux ;
- Sa clarté ;
- La mise à jour est facile ;
- La formation des programmeurs est peu exigeante.

❖ **Inconvénients des règles de productions :**

- L'inefficacité : il est notamment impossible de prévoir un déroulement efficace pour une séquence d'action, et le balayage systématique des règles génère un travail de recherche considérable dès que le volume de la base devient significatif ;
- Mauvaise lisibilité du raisonnement et de la cohérence de la base : l'utilisateur manque de la vue d'ensemble sur la connaissance d'un problème qui permet de contrôler sa cohérence et de percevoir son organisation logique.

### II.3. Réseaux sémantiques :

Les réseaux sémantiques permettent la représentation des connaissances sous forme de graphes composés de nœuds reliés par des arcs.

- Les nœuds représentent les objets (concepts, événements, situations) ;
- Les arcs traduisent les relations ou les liens sémantiques entre ces objets.

**Exemple :**



**Figure 12: Réseau sémantique simple.**

L'interprétation de ce schéma peut se faire aisément comme suit :

- Un pigeon est un oiseau ;
- Un oiseau possède des ailes.

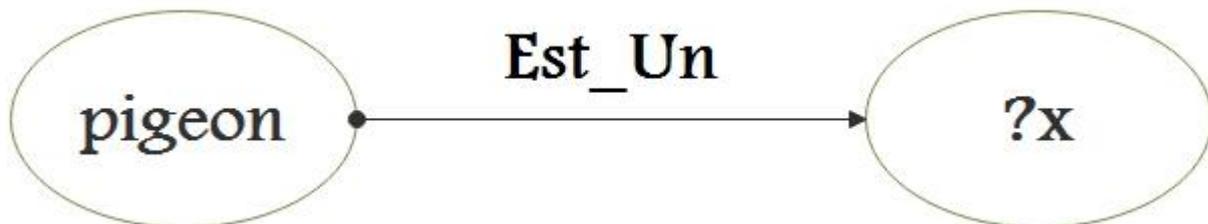
Par conséquent, on peut dériver qu'un pigeon possède des ailes.

Les réseaux sémantiques englobent deux techniques de raisonnement :

- **L'héritage** : Transmission des propriétés des concepts les plus élevés vers leurs fils.
- **Le filtrage** : Dans un réseau sémantique, une question est traduite sous forme d'une structure objet-relation inconnu (information recherchée). Le filtrage consiste à mettre en correspondance la structure ainsi obtenue avec le réseau initial en tenant compte des nœuds et arcs. Si une telle correspondance peut être établie alors la réponse à la question est obtenue.

**Exemple :**

Un exemple de raisonnement par réseau sémantique est donné dans le schéma suivant. Il consiste à représenter par un réseau le contenu d'une requête ou d'un but, qui est ensuite mis en correspondance avec la base de faits, en utilisant les techniques de filtrage.



**Figure 13: Représentation d'une requête par réseau sémantique.**

Le filtrage nous donnera la réponse : ?x= oiseau.

❖ **Avantages des réseaux sémantiques :**

- Les réseaux sémantiques sont faciles à lire, à comprendre du fait de l'organisation des connaissances permise par le mécanisme d'héritage. De plus, l'emploi du "pattern-matching" comme mécanisme de raisonnement apporte une facilité d'utilisation.
- Ils sont capables de gérer correctement les exceptions.

❖ **Inconvénients des réseaux sémantiques :**

- Il manque une sémantique formelle et une terminologie standard aux réseaux sémantiques. Certains types de connaissances (procédurales ou quantifiées) ne peuvent pas être représentés. Par exemple, ce modèle est incapable de

traiter la quantification mis en jeu dans la phrase : "tout parent aime ses propres enfants".

- Du fait de la simplicité structurelle des nœuds, un réseau sémantique est complexe dès qu'il contient beaucoup d'informations. Il est ainsi d'une manipulation délicate et difficile à étendre ou à modifier. La modularité apparente n'est pas un réel avantage.
- il n'existe pas encore de procédures qui manipulent rapidement ces réseaux.
- Enfin, les systèmes qui utilisent ce formalisme ont des difficultés pour fournir des explications sur leur raisonnement.

#### II.4. Représentation orienté objet :

Dans la conception par objet, on ne s'intéresse pas à ce que fait le système, mais à quoi il le fait. On recense pour cela les objets.

Aujourd'hui, les langages orientés-objets connaissent un succès incontestable en intelligence artificielle où ils sont considérés comme outils de représentation de connaissances.

La notion d'objet possède trois modèles différents :

- **Génie logiciel** : langage de classes.
- **Représentation de connaissances en psychologie cognitive** : langage de frames et scripts.
- **Traitement parallèle** : langage d'acteur.

##### II.4.A. Langage de classes : [11]

Ils ont apparus comme une solution à des problèmes en génie logiciel, ces langages sont caractérisés par :

###### ➤ **Concept d'objet et classe :**

Une classe d'objets est un modèle définissant une famille d'objets ayant des propriétés analogues. La classe d'objets décrit ces propriétés en termes d'attributs et de méthodes.

Un objet est créé par instanciation d'une classe, c'est-à-dire par création d'un élément conforme au modèle que constitue sa classe. Un objet est considéré comme un élément autonome et actif. Il a les propriétés et les comportements (méthodes) définis au niveau de sa classe. Dans ce cas on utilise également le terme d'instance.

**Remarque :** il ne faut pas confondre entre classe et objet : le premier désigne un ensemble d'éléments (un ensemble de clients par exemple), et le second désigne un élément particulier de cet ensemble (un client particulier). En d'autres termes : une classe est un type, un objet est une occurrence du type.

###### ➤ **Méthode, attribut et message :**

Chaque objet réagit aux messages qu'il reçoit, et émet lui-même des messages vers d'autres objets.

Dans la plupart des langages à objets, l'envoi de message correspond à un appel de fonction (méthode).

Une méthode définit les informations (les paramètres) nécessaires pour chacun des messages et décrit les traitements réalisés par l'objet lors de la réception d'un message correspondant.

Les objets possèdent des valeurs qui leurs sont propres : les attributs.

➤ **L'encapsulation :**

Les objets gèrent totalement les valeurs de leurs attributs, qu'eux seuls peuvent modifier directement grâce à leurs méthodes. On dit que les attributs sont encapsulés (cachés), c'est-à-dire qu'ils ne peuvent être initialisés ou consultés que par les méthodes de la classe.

➤ **L'héritage :**

Il permet de construire des hiérarchies de classes dans lesquelles une classe possède les attributs et les méthodes de la classe mère (**super classe**).

On dit qu'une classe A hérite d'une classe B si elle (A) possède toutes les caractéristiques de celle-ci (B) en plus des siennes.

On distingue deux types d'héritage :

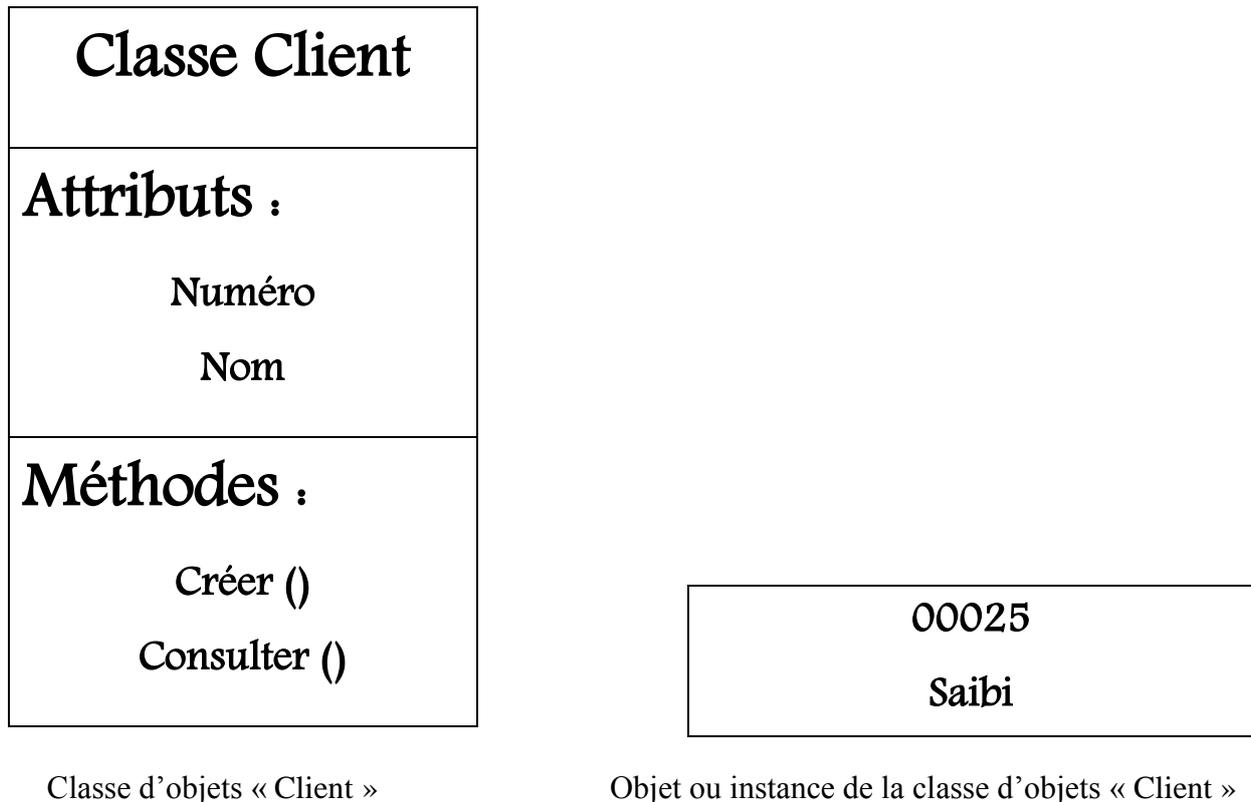
- **Simple** : une sous-classe ne possède qu'une seule classe parente directe.
- **Multiple** : une classe peut avoir plusieurs classes parentes directes.

➤ **Le polymorphisme :**

Le polymorphisme permet de donner le même nom aux méthodes appartenant à des classes d'objets différentes.

Il est conséquence du principe d'héritage, car il permet d'attribuer un nom à une méthode qui sera utilisée dans toute la hiérarchie de cette classe.

Exemple :



**Figure 14: Classe d'objets et objet.**

#### II.4.B. Langage de frames et de scripts : [7]

Bien que développés indépendamment au début des années 70, les frames (Schank et Abelson 1977) et les scripts (Minsky 1975) ont beaucoup de points en commun et sont développés dans un même document. L'idée clef des frames et des scripts, empruntée aux travaux en psychologie cognitive, en particulier en ce concerne l'acquisition de connaissances, est que notre connaissance et notre perception des concepts, des événements et des situations que nous utilisons, sont organisées autour de contenus prototypiques et de caractéristiques particulières que nous attendons relativement à ces concepts, événements et situations.

##### Les frames :

Un frame est une unité de connaissances (structure de donnée ou prototype) décrivant une situation (ou un objet) en regroupant un ensemble d'éléments de description relatifs à cette situation (ou à cet objet) appelés attributs qui sont eux-mêmes décrits par un ensemble de facettes.

La forme syntaxique d'un frame :

(Nom-frame

```

      (Attribut 1  (Facette 1)
                    (Facette 2)
                    .
                    .
                    .
                    (Facette i))
      .
      .
      .
      (Attribut n  (Facette 1)
                    (Facette 2)
                    .
                    .
                    .
                    (Facette j)))

```

Il existe deux types de facettes :

- **Facettes déclaratives** : elles permettent d'associer des valeurs aux attributs par défaut ou un ensemble de valeurs possibles pour un attribut.
- **Facette procédurales** : elles spécifient des procédures (réflexes) qui seront exécutées à chaque fois que la valeur d'un attribut change ou lors de la demande d'une valeur absente pour un attribut ou lors d'une suppression d'une valeur d'un attribut.

Dans une structure à base de frames, on distingue :

- Les prototypes servant à définir toute une classe d'objets et devant être considérés comme des « moules » décrivant la structure des objets (instances) qu'ils engendrent.
- Les instances, réalisations particulières des prototypes.

L'attribut « Sorte-de » établit une relation d'héritage entre un frame et un sous-frame qui est sa spécialisation. Un sous-frame est une spécialisation d'un ou plusieurs frames pères appelés super-frames dont il hérite les couples Attribut-Facette, cet héritage est dynamique.

**Exemple :**

<b>Frame Chaise</b>	
Sorte-de	: Meuble
Nombre-de-pieds	: <b>doit être</b> entier <b>Par défaut</b> 4
Style-du dossier	: <b>doit être</b> Droit, Rembourré
Nombre-de-bras	: <b>doit être</b> 0, 1 ou 2

<b>Frame Chaise-de-Paul</b>	
Sorte-de	: Chaise
Nombre-de-pieds	: 4
Style-du dossier	: Rembourré
Nombre-de-bras	: 0

**Remarque :**

- Le frame « Chaise » décrit une catégorie, qui est un sous-ensemble de la catégorie plus générale « Meuble ».
- Le frame « Chaise-de-Paul » décrit un objet particulier, unique, qui est une instance de la catégorie « chaise ».
- Les facettes déclaratives qui expriment les valeurs par défaut sont précédées par le mot-clé « **par défaut** », et les facettes déclaratives précédées par le mot-clé « **doit être** » spécifient le domaine de définition.

**✚ Les scripts :**

La notion de scripts (scénario) a été introduite, sur le modèle des frames, pour décrire, non plus des objets, mais des scènes de la vie courante.

Un script est une séquence d'actions prédéterminée et stéréotypée qui définit une situation connue. Il a une structure analogue aux frames.

**Exemple :**

Soit le script qui décrit le fait d'aller au restaurant, ceci comprend typiquement l'entrée dans le restaurant, la recherche d'une table, la commande, le repas lui-même, la demande d'addition, le paiement et le départ du restaurant.

```
SCRIPT " manger- au- restaurant"
```

```
ELEMENTS: (restaurant, agent, nourriture, menu, tables, chaises)
```

```
ROLES: (clients, serveur, chefs)
```

```
POINT- DE- VUE: clients
```

```
MOMENT : (heure d'ouverture du restaurant)
```

```
LIEU : (emplacement du restaurant)
```

```
SCÉNARIO:
```

D'abord	: script « Entrer restaurant »
Puis	: script « Attirer -l'attention -du -client- de- restaurant »
Puis	: script « Prendre- place- à- table »
Puis	: script « Passer- commande »
Puis	: script « Manger » sauf si (longue attente) alors script « Sortie- en- colère »
Puis	: si qualité nourriture >convenable) alors script « Féliciter- le chef »
Puis	: script « Payer- l'addition »
Enfin	: script « Quitter-restaurant ».

#### II.4.C. Langage d'acteurs :

C'est un modèle de représentation des connaissances qui est issu des travaux de recherches sur le traitement parallèle. Il contient presque toutes les notions du principe objet à part celle de l'héritage.

- **L'acteur** : Il s'agit d'un objet actif, et autonome communiquant avec ses semblables en envoyant des messages asynchrones et unidirectionnels. Il englobe au sein d'une même entité :
  - Les données locales (accointances) qui sont les autres que l'acteur connaît directement.
  - Un comportement qui définit les actions que l'acteur entreprend au cours de son existence.
- **L'envoi des messages** : Consiste à établir une communication en expédiant un message d'un acteur à un autre, par l'intermédiaire d'un acteur message (mandataire).

### III. Principes du fonctionnement d'un moteur d'inférence dans un système de production :

Tout moteur d'inférence est caractérisé par :

- Un cycle de base ;
- Un mode d'invocation des règles ;
- Une stratégie de recherche ;
- Une stratégie de contrôle.

#### III.1. Le cycle de base d'un moteur d'inférence : [9]

Le cycle de base d'un moteur d'inférence est composé de deux phases :

- la phase d'évaluation ;
- la phase d'exécution.

**III.1.A. La phase d'évaluation :**

Le moteur d'inférence détermine s'il existe dans la base de règle courante des règles à déclencher au vu de l'état courant de la base de fait et si oui, quelles sont elles ? Cette phase comprend trois étapes :

- La restriction ;
- Le filtrage ;
- La résolution des conflits.

- **La restriction :**

Elle a pour objectif de restreindre l'ensemble des faits (BF) et des règles (BR) un sous ensemble (F1) de BF et un sous ensemble (R1) de BR qui a priori d'être comparés lors de l'étape suivante.

- **Le filtrage :**

Dans cette étape, le moteur d'inférence compare la partie prémisse (déclencheur) de chacune des règles de R1 par rapport à l'ensemble F1 de fait afin de déterminer un sous ensemble R2 de R1.

- **La résolution des conflits :**

Le moteur d'inférence détermine les règles qui doivent être effectivement déclenchées par un sous ensemble R3 et R2, si R2 est vide il n'aura pas d'exécution pour le présent cycle.

**III.1.B. La phase d'exécution :**

Elle consiste à activer la règle sélectionnée lors de l'étape précédente (R3 bien sur s'il n'est pas vide). L'activation de la règle consiste à rajouter sa conclusion (partie action) dans la base des faits.

**Remarque :**

Le cycle « évaluation-exécution » est répété par le moteur d'inférence jusqu'à ce le but posé soit atteint ou jusqu'à ce qu'il n'y ait plus de règles applicables (activable).

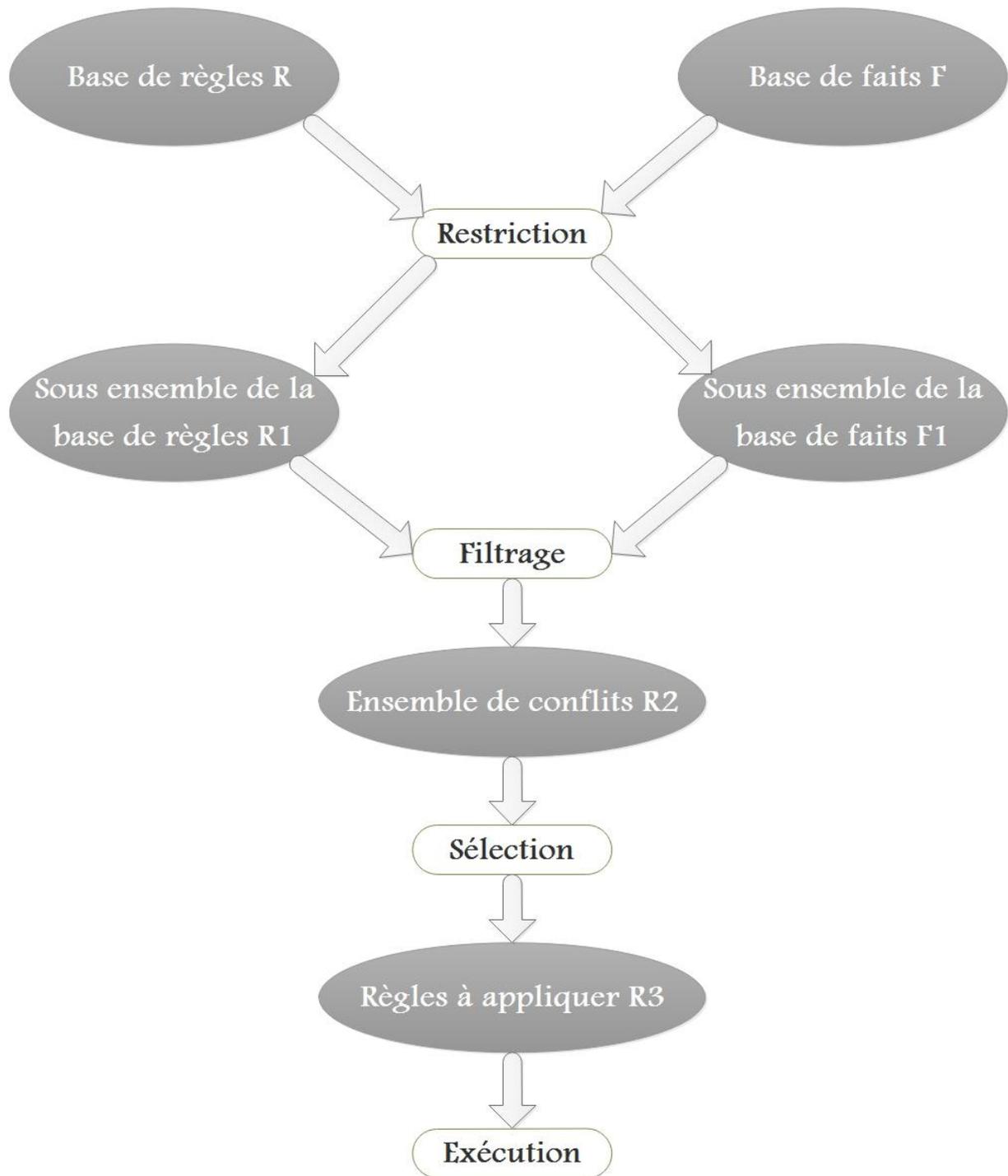


Figure 15: Cycle de base d'un moteur d'inférence. [9]

### III.2. Le mode d'invocation des règles : [9]

On entend par « invocation des règles » le chaînage des règles suivant un mode de raisonnement précis pour aboutir à un but particulier et enrichir sa base de connaissances.

Le moteur d'inférences fonctionne selon l'un des trois modes de chaînage suivants :

- Le chaînage avant ;
- Le chaînage arrière ;
- Le chaînage mixte.

**III.2.A. Le chaînage avant :**

Dans le mode de chaînage avant (connu aussi comme le mode guidé par les données), le moteur d'inférences ne sélectionne que les règles dont les conditions (prémises) sont satisfaites au vu de l'état de la base de faits pour arriver au but recherché (un fait qui répond à la question posée), puis applique une de ces règles afin d'ajouter d'autres faits à la base.

Cet enchaînement est réitéré jusqu'à ce que plus aucun fait nouveau ne puisse être déduit (on parle alors de saturation) ou que le but soit atteint.

**III.2.B. Le chaînage arrière :**

Un moteur d'inférence qui suit le mode de chaînage arrière, part du but et essaie de remonter aux faits pour le prouver (parcours guidé par le but). En effet, le moteur sélectionne les règles dont la partie action (ou conclusion) correspond au but recherché.

Les prémisses/conditions de ces règles deviennent elles aussi des sous buts à prouver et ainsi de suite. Cet enchaînement s'arrête lorsque tous les sous buts sont prouvés le but est alors lui aussi prouvé ou lorsqu'il n'est plus possible de sélectionner des règles.

**III.2.C. Le chaînage mixte :**

Le chaînage mixte est une combinaison du chaînage avant et du chaînage arrière.

Un moteur qui suit ce mode opère généralement comme suit :

Tant que des règles sont applicables (conditions satisfaites), il procède par chaînage avant jusqu'à satisfaction du but ou jusqu'à saturation de la base de faits ;

Si le but n'est toujours pas prouvé, il choisit alors une règle dont la partie conclusion correspond au but et essaie de prouver ses prémisses inconnues par chaînage arrière en veillant à modifier la base de faits à chaque nouveau sous but prouvé ;

Si le but n'est toujours pas prouvé, il repart en chaînage avant en tenant compte des nouveaux faits prouvés et éventuellement en demandant de nouveaux faits à l'utilisateur et ainsi de suite jusqu'à ce que le but soit prouvé ou jusqu'à ce qu'aucune règle ne soit applicable.

**III.3. La stratégie de recherche : [11]**

Dans un moteur d'inférence, en plus des modes d'invocation des règles (méthodes de chaînage), on superpose toujours une stratégie de recherche qui lui permet de mettre en œuvre un enchaînement des règles lors de la recherche de la solution.

Il existe quatre stratégies de recherche principales :

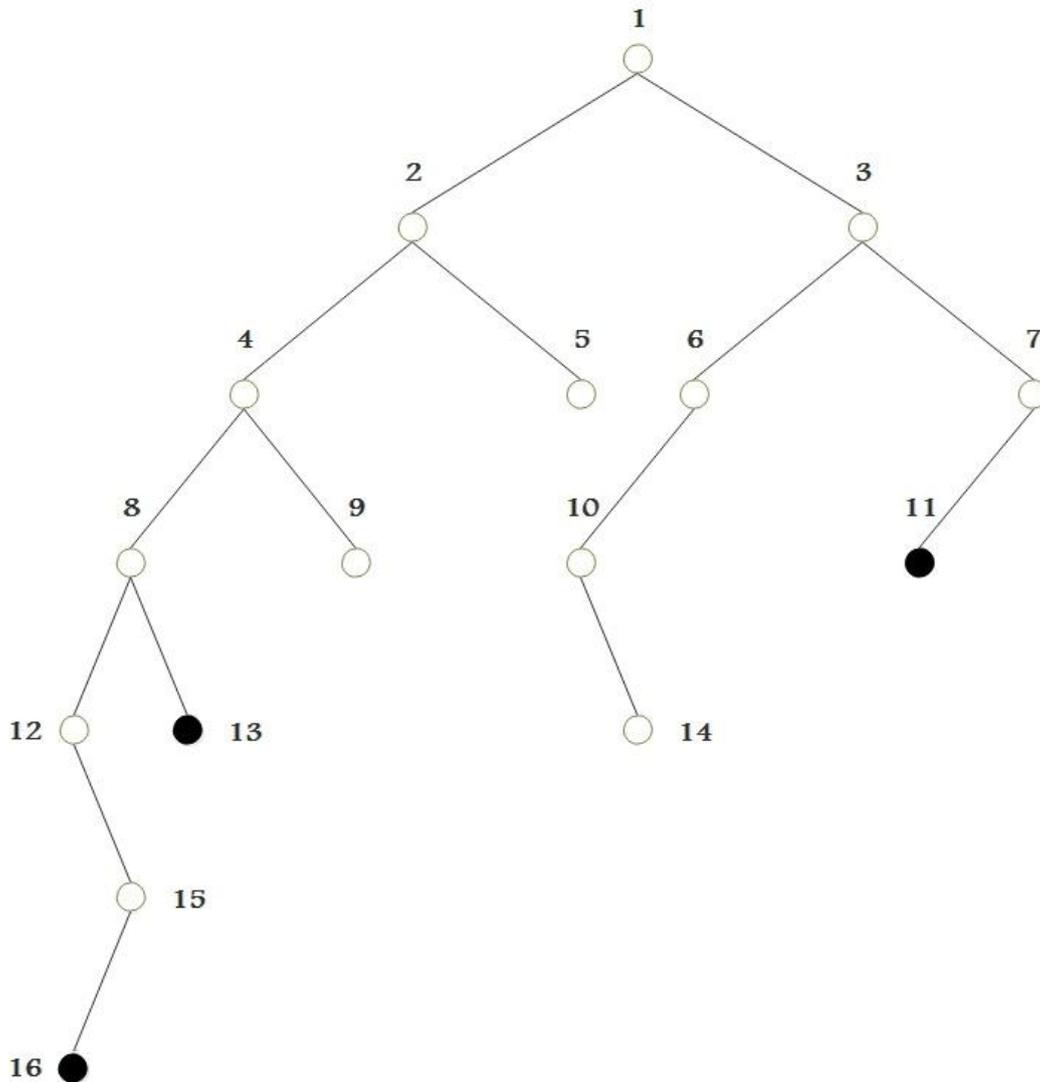
- La recherche en largeur d'abord ;
- La recherche en profondeur d'abord ;
- La recherche en profondeur limitée ;
- La recherche heuristique.

**III.3.A. La recherche en largeur d'abord :**

Cette stratégie de recherche permet de parcourir l'arbre à partir de la racine de gauche à droite.

**Exemple :**

Considérons l'arbre de la figure suivante, les nœuds buts (les nœuds solutions) sont représentés en noir.



**Figure 16: Recherche en largeur d'abord.**

La méthode de recherche mise en œuvre va s'exécuter jusqu'à la rencontre d'un nœud solution. Dans l'exemple ci-dessous, la recherche s'arrêtera après 11 itérations. Les chiffres à devant les nœuds correspondent à l'ordre de prise en considération.

**III.3.B. La recherche en profondeur d'abord :**

Cette stratégie de recherche permet de parcourir l'arbre à partir de la racine en explorant une branche jusqu'à son terme avant de passer à la branche adjacente.

**Exemple :**

Considérons l'arbre de la figure suivante, les nœuds buts (les nœuds solutions) sont également représentés ici en noir.

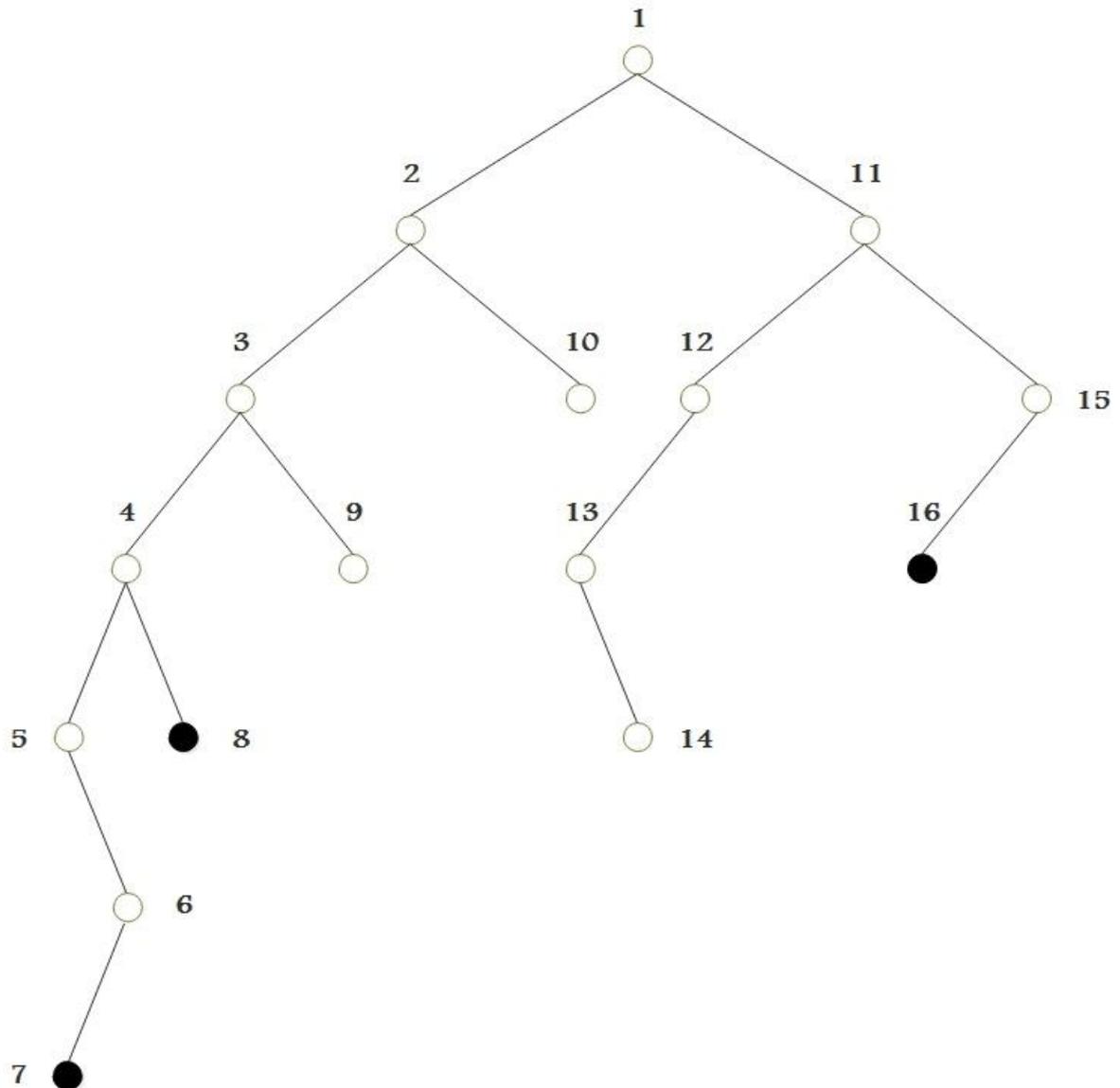


Figure 17: Recherche en profondeur d'abord.

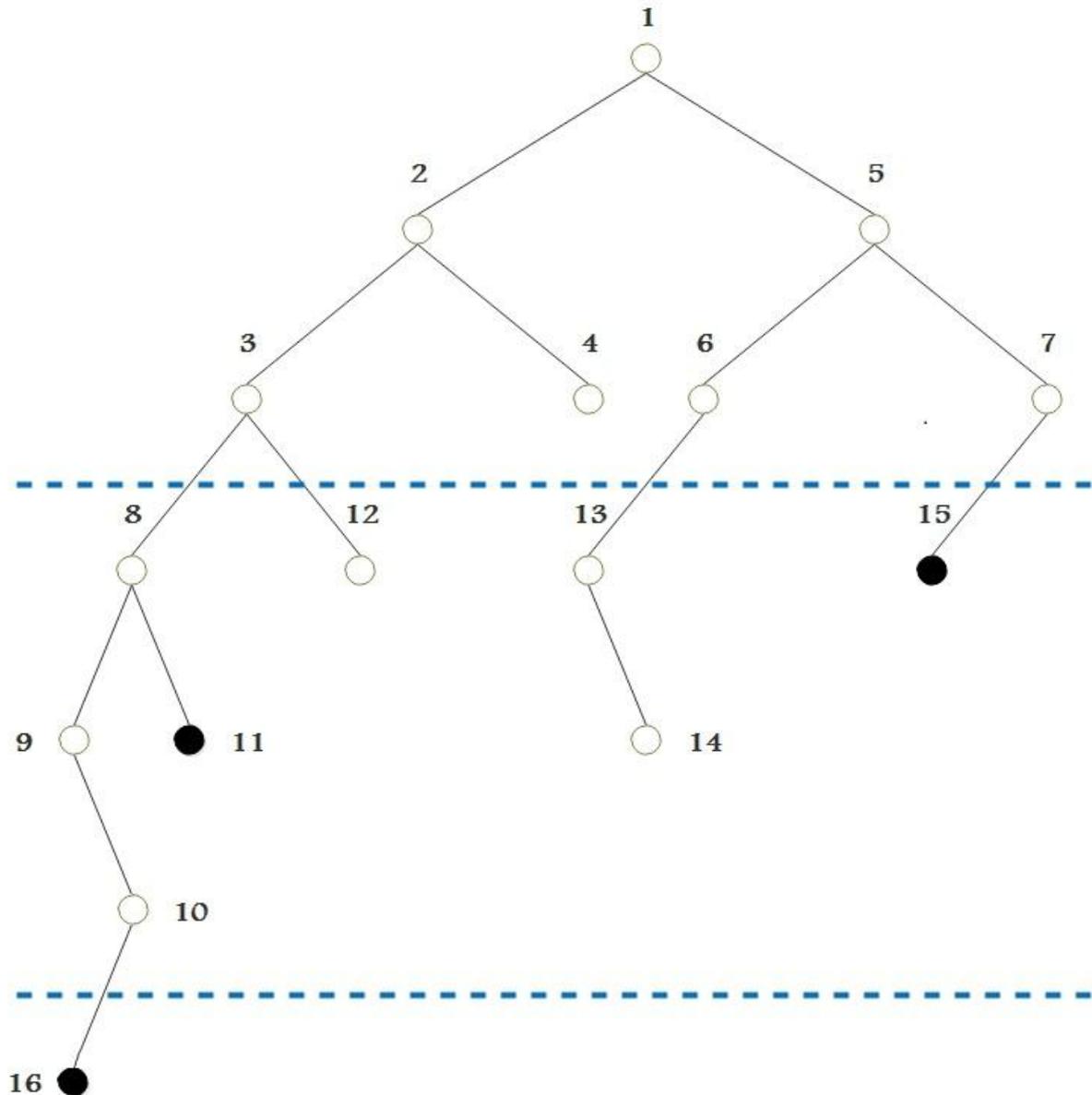
La méthode de recherche mise en œuvre va s'exécuter jusqu'à la rencontre d'un nœud solution. Dans l'exemple précédent, la recherche s'arrêtera après 7 itérations.

Dans le cas où les nœuds solutions se trouvent à droite, le temps de recherche est alors maximum.

### III.3.C. La recherche en profondeur limitée :

Cette stratégie de recherche est un mélange des deux approches précédentes. Son objectif principal est de réduire les temps de recherche de solutions. Dans ce cas, on découpe l'arbre en tranches de  $k$  niveaux. On effectue ensuite une recherche en profondeur dans chaque tranche avant de passer à la tranche suivante.

**Exemple :** considérons l'arbre de la figure suivante où  $k$  est fixé à 3 :



**Figure 18: Recherche en profondeur limitée.**

Dans l'exemple précédent la recherche s'arrêtera après 11 itérations.

### III.3.D. Les recherches heuristiques :

Chacune des trois méthodes présentées précédemment possède un inconvénient majeur : leur efficacité dépend de la configuration initiale qui est présentée. De plus, il s'agit de stratégies de recherche exhaustives, c'est-à-dire qu'on explore l'ensemble des solutions possibles.

Il existe d'autres méthodes, dites heuristiques, pour lesquelles on n'explore pas ensemble des solutions. En effet, plutôt que de procéder à des recherches aveugles, on exploite la connaissance spécifique sur le problème traité. Ceci se traduit généralement par

l'ajout d'une fonction d'évaluation dont l'objectif est de mesurer le chemin le plus intéressant menant au but recherché.

### **III.4. La stratégie de contrôle :**

On distingue les stratégies suivantes :

- La monotonie ;
- La non monotonie ;
- L'hypothèse du monde clos et du monde ouvert ;
- Le régime de contrôle par tentative.

#### **III.4.A. La monotonie :**

Un système expert est dit monotone si lors du raisonnement, aucune connaissance ne peut être retirée de la base, et aucune connaissance ajoutée à la base n'introduit de contradiction.

#### **III.4.B. La non monotonie :**

Un système expert fonctionne de façon non monotone s'il peut modifier ou retirer des faits précédemment établis, ou bien si on leur associe un langage qui offre des primitives qui permettent de supprimer, ou ajouter des connaissances lors du raisonnement.

#### **III.4.C. L'hypothèse du monde clos et du monde ouvert :**

L'hypothèse du monde clos, toute assertion non explicitement présente dans la base de connaissances est considérée comme fausse, d'où l'absence d'un fait est équivalente à sa négation.

Le monde ouvert, contrairement à celle du monde clos, l'hypothèse du monde ouvert stipule que toute connaissance doit être explicitement représentée dans la base de faits. Ainsi, la négation d'un fait n'est vérifiée que si elle figure dans la base de faits.

#### **III.4.D. Le régime de contrôle par tentative :**

Ici, en cas d'échec, tous les faits établis dans le raisonnement sont retirés de la base de faits, et cette dernière retrouve son contenu avant l'échec. C'est-à-dire, le moteur d'inférence reconsidère l'ensemble de conflits du cycle antérieur et examine la possibilité de déclencher d'autres règles.

Lorsqu'il y'a remise en cause de l'application d'un ensemble de règles, on dit alors que le moteur opère avec un retour arrière.

## **IV. Conclusion :**

L'IA propose des alternatives intéressantes aux problèmes d'aide à la décision, et c'est là que s'impose l'approche Systèmes Experts. Les systèmes experts ne sont autres que des programmes informatiques intelligents dotés d'une grande masse de connaissances, écrits dans un style particulier mettant en évidence, la distinction entre connaissance et raisonnement.

Dans ce chapitre, nous avons passé en revue les systèmes experts, leur fonctionnement ainsi que les différents formalismes utilisés pour représenter les connaissances qu'ils exploitent.

Après avoir vu d'une façon globale le champ d'application, et après avoir retenu une idée globale sur les stratégies de fonctionnement des systèmes experts, le chapitre qui succède est réservé pour la conception de notre système, en optant le système expert approprié à partir de son domaine d'application.

*CHAPITRE* III

---

***ANALYSE  
ET  
CONCEPTION***

**Introduction :**

Après avoir retenu une idée globale sur les systèmes experts, ainsi que le domaine d'expertise (évaluation des dommages poste-sismiques), nous nous intéressons dans le présent chapitre à l'analyse et à la conception de notre système qu'on nommera « EDPS ».

Rappelons que l'objectif de notre travail est de concevoir et de mettre en place un S.E pour la gestion de crises après événement sismique majeur. Notre système offrira principalement un module d'acquisition des connaissances (ces dernières sont basées sur la logique d'ordre 1), et un moteur d'inférence capable de raisonner sur ces connaissances.

Ce chapitre est organisé en deux sections :

- La première section portera sur l'analyse de notre système ;
- La seconde section est dédiée à la conception de ce dernier.

**I. Analyse :****I.1. Choix du domaine et objectif du système :**

Après chaque séisme, une mission constituée d'ingénieurs en génie civil est chargée de l'expertise et l'évaluation des constructions. Il s'agit d'inventorier les parties du patrimoine utilisable immédiatement, celles qui nécessitent des réparations et celles qui sont irrécupérables et ainsi répondre rapidement aux besoins des autorités sur le sort à réserver aux ouvrages touchés par le sinistre sur le plan technique et leur permettre de connaître l'ampleur des dégâts afin de mobiliser les moyens financiers, matériels et humains pour parer ce sinistre.

Une fiche technique regroupant une cinquantaine d'informations est utilisée par l'équipe chargée de l'expertise des constructions. Cette fiche comporte des renseignements d'ordre général (identification, usage, nombre de niveaux, ...), des observations concernant les problèmes de sols, des évaluations partielles des dommages subis par les éléments structurels et par les éléments secondaires constituant la construction, des commentaires sur les causes probables des dommages pour aboutir à une évaluation globale des dommages subis par la construction.

L'objectif de notre système est d'arriver à donner une évaluation globale d'une construction à partir des évaluations partielles concernant les différents éléments constituant la construction ainsi que d'un certain nombre de renseignements et d'observations concernant celle-ci.

**I.2. Intérêts du système EDPS :**

Notre système pourra être :

- Un outil de travail utilisé par la personne chargée par les autorités d'expertise des constructions l'aidant dans sa prise de décision en rendant l'évaluation finale indépendante du facteur humain (minimiser l'erreur humaine) ;
- Un outil pédagogique pour les ingénieurs du génie civil du moment que le système traite un domaine qui n'est pas abordé durant leur cursus d'études universitaires et pour lequel il pourrait être mobilisé par les autorités locales ou nationales ;

- Cet outil servira aussi d'un moyen pour répondre aux besoins du grand public (toute personne n'ayant pas des connaissances sur le domaine d'expertise), à savoir, il permettra à ce dernier d'avoir une idée d'ordre général sur l'état de son bien.

### **I.3. Les utilisateurs de notre système :**

Comme tout système expert, EDPS s'adresse à deux catégories d'utilisateurs :

#### **I.3.A. Les utilisateurs ordinaires :**

Ce sont les consommateurs du logiciel qui posent leurs problèmes au système et attendent une réponse de celui-ci.

Ils sont classés en deux sous-catégories :

- Les propriétaires (les citoyens, ou bien le grand public) ;
- Les experts (ce sont les personnes chargées par les autorités d'expertise des constructions).

Cette catégorie d'utilisateur est décrite en détail dans la section précédente (**I.2**).

#### **I.3.B. Les administrateurs :**

Ce sont les experts en pathologie des constructions et du génie parasismique qui sont habilités à changer la base de connaissances du système soit en intégrant de nouvelles données, soit en modifiant celles qui existent déjà ou en supprimant celles qui seront obsolètes.

### **I.4. Les intervenants du projet :**

Les différents acteurs de notre projet sont :

- Un coordinateur qui fera le suivi de notre travail, il s'agit de notre promoteur ;
- Un expert du domaine d'expertise qui est un doctorant au département de génie civil de l'UMMTO.
- Nous avons joué le rôle du cogniticien durant la première phase pour objectif l'extraction des informations nécessaires auprès de l'expert et leur expression dans un formalisme de représentation des connaissances adéquat. Durant la deuxième phase, c'est le rôle de l'informaticien chargé des aspects informatiques et de la conception du système expert que nous avons tenu.

### **I.5. Extraction de la connaissance :**

Nous nous sommes basé sur la technique d'interview pour l'extraction des connaissances qui était un vrai goulot d'étranglement et sur laquelle repose toute la puissance et l'efficacité du système.

Ainsi, plusieurs séances de travail et de discussions se sont déroulées avec l'expert qui nous a expliqué les aspects pratiques du diagnostic des constructions ainsi que son raisonnement devant des cas pratiques tirés de son expérience. Dans un premier temps, nous étions confrontés à un jargon auquel il a bien fallu se familiariser en se servant sur quelques livres et revues portant sur les constructions et effets des séismes sur elles.

## II. Conception :

### II.1. Conception globale :

Pour réaliser un système expert dans n'importe quel domaine il faut trouver le savoir faire (l'expertise et donc les experts). C'est la tâche la plus difficile que de répertorier et classer les différentes règles. Dans le cadre de notre travail, nous avons pris attache avec un doctorant en génie civil qui a alimenté notre base de connaissances.

« EDPS » est un système expert dédié à l'évaluation des dommages et post-sismiques, il est doté de différents modules et interfaces à savoir :

- Une base de connaissances (BC) :
  - o Une base de faits (BF);
  - o Une base de règles (BR);
- Un moteur d'inférence (MI) ;
- Un module d'acquisition de connaissances (MAC) ;
- Les modules d'interfaces :
  - o Interface propriétaire ;
  - o Interface expert ;
  - o Interface administrateur ;
- Un module d'aide (MA) ;
- Un module d'explication (ME).

#### II.1.A. La base de connaissances (BC) :

La connaissance modélisée est stockée dans une base de faits et une base de règles en utilisant la logique des prédicats (logique d'ordre 1) et les règles de productions.

- **La base de faits (BF) :**

Suivant les connaissances collectées, nous avons choisis la logique des prédicats pour les représenter.

Un fait est donc représenté sous la forme suivante :

Fait = prédicat (argument).

- o Prédicat : c'est une fonction booléenne, qui prend un ou plusieurs arguments.
- o Argument : peut être un prédicat, une valeur alphabétique ou numérique (constante) ou une variable.

- **La base de règles (BR) :**

Elle rassemble les connaissances et le savoir faire de l'expert du domaine, ces connaissances sont représentées par des règles de productions dont la syntaxe est la suivantes :

SI <conditions> ALORS <conclusion>

- Condition : est un fait ou plusieurs faits, c'est la partie déclencheur de la règle.
- Conclusion : est un fait, c'est la partie déclenchée (déduite) de la règle.

### **II.1.B. Le moteur d'inférence (MI) :**

Le moteur d'inférence utilise la base de règles pour répondre à la demande des utilisateurs de notre système, il représente le noyau du programme qui s'occupe de parcourir la base de connaissance à la recherche de résultats pour les différentes requêtes.

### **II.1.C. Le module d'acquisition des connaissances (MAC) :**

Un système expert devant être évolutif et extensible, le module d'acquisition de connaissances permettra l'ajout, la modification et la suppression d'une ou plusieurs règles.

Chaque règle à ajouter à la BR subit un contrôle de cohérence qui consiste en deux tests :

- Test de redondance : vérifier l'inexistence dans la BR d'une règle équivalente afin d'éviter la redondance.
- Test de contradiction : vérifier l'inexistence dans la BR d'une règle ayant les mêmes prémisses mais une conclusion différente.

### **II.1.D. Les modules d'interfaces :**

Étant donné que les systèmes experts sont des logiciels interactifs, nécessitant un dialogue homme/machine, cela rend l'utilisation des interfaces indispensable.

Comme notre système est manipulé par trois types d'utilisateurs, nous le dotons de trois interfaces :

- **L'interface Propriétaire :**

Le propriétaire est dirigé par cette interface qui lui permettra de mieux décrire le problème à diagnostiquer et de récupérer le résultat du diagnostic ainsi que les explications demandées.

- **L'interface Expert :**

Cette interface permet à l'expert (toute personne chargée par les autorités d'expertise des constructions) de dialoguer avec le système pour qu'il puisse introduire les faits et obtenir le résultat du diagnostique.

- **L'interface Administrateur :**

Reliée au module d'acquisition de connaissances, cette interface protégée permettra uniquement à l'expert ou au cogniticien qui sont agrès par le CGS ou bien le CTC.

Ces derniers sont aptes à manipuler la base de données et la base de règles, c'est-à-dire :

- Mettre à jour la base de règles ;
- Ajouter, modifier et supprimer un expert (c'est-à-dire sélectionner les différents experts compétents capables d'évaluer les biens).

**II.1.E. Le module d'aide (MA) :**

Notre système est destiné au grand public et parmi eux des non informaticiens et des non connaisseurs en pathologie des constructions et en risque post-sismiques.

Ce module leur permet une utilisation facile du système en fournissant des explications sur le système (Logiciel) : sa structure et son fonctionnement.

Ainsi il permettra de guider l'utilisateur (l'expert et le propriétaire) lors de l'utilisation du système EDPS.

**II.1.F. Le module d'explication (ME) :**

Comme son nom l'indique, la fonction de ce module est d'expliquer le raisonnement aux utilisateurs d'EDPS après l'élaboration d'un diagnostic et à la demande de l'utilisateur en affichant toutes les règles déclenchées au cours du diagnostic accompagnées de leurs commentaires.

Le schéma ci-dessous illustre l'architecture détaillée du système EDPS :

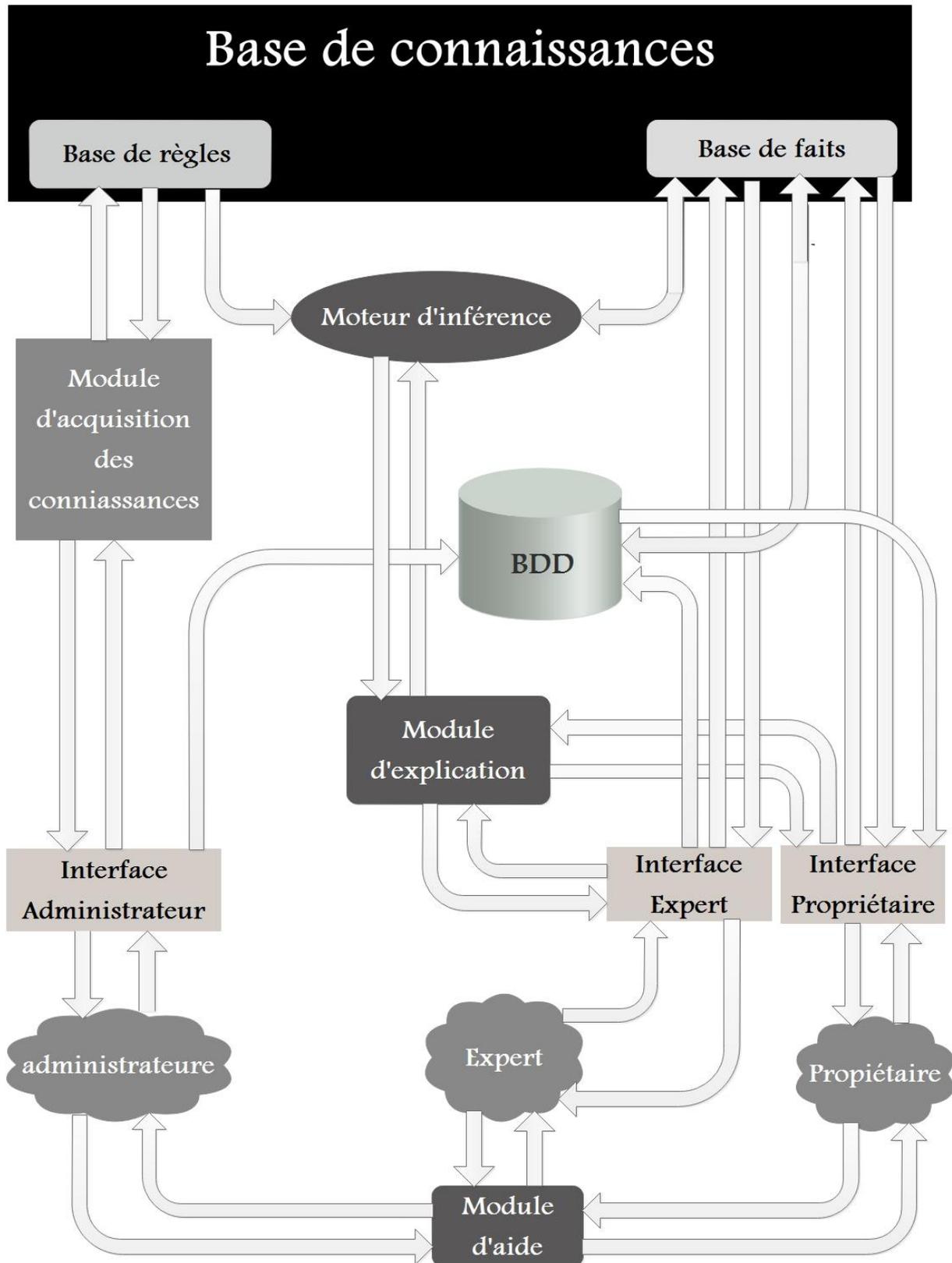


Figure 19: Architecture détaillée du système EDPS.

## II.2. Conception détaillée :

L'objectif de cette partie est de fournir une étude conceptuelle suffisamment détaillée de notre travail. Pour cela, nous avons opté pour la modélisation des différents concepts du domaine d'application, afin de faciliter la compréhension, et la conception de notre système.

Ainsi cette phase sera divisée en deux :

- Phase 1 : Conception de notre base de connaissance à l'aide d'un langage basé sur la logique du premier ordre.
- Phase 2 : Conception des modules et les interfaces graphiques à l'aide d'un langage orienté objet, ainsi nous nous basons dans sa conception, sur le langage UML.

### II.2.A. La conception logique :

Pour concevoir notre base de connaissances nous avons opté sur la modélisation basée sur la logique des prédicats.

Rappelons qu'un prédicat contient des quantificateurs et des variables, permettant de vérifier l'existence d'une propriété pour un objet inconnu.

Cette propriété est soit :

- Vraie pour tout objet :  $\forall X p(X)$ .
- Vraie pour certains objets :  $\exists X p(X)$ .

- **La base de faits :**

La base de faits constitue la mémoire de travail du système, elle contient des de la forme suivante :

Prédicat (variable).

Donc une instance d'un fait est un : prédicat (valeur).

Les faits conçus pour notre S.E sont représentés dans le tableau suivant :

Fait	Désignation
dep(DEP)	Le niveau de dommage de l'élément porteur (dep) est de valeur (DEP)
dec(DEC)	Le niveau de dommage de l'élément de contreventement (dec) est de valeur (DEC)
deptt(DEPTT)	Le niveau de dommage de l'élément plancher-toiture terrasse (deptt) est de valeur (DEPTT)
deti(DETI)	Le niveau de dommage de l'élément toiture inclinée (deti) est de valeur (DETI)

dees(DEES)	Le niveau de dommage de l'élément escalier (dees) est de valeur (DEES)
dere(DERE)	Le niveau de dommage de l'élément remplissage extérieur (dere) est de valeur (DERE)
dei(DEI)	Le niveau de dommage de l'élément intérieur (dei) est de valeur (DEI)
deex(DEEX)	Le niveau de dommage des autres éléments extérieurs (deex) est de valeur (DEEX)

**Tableau III.1 : Tableau descriptif des faits conçus dans notre système.**

**Remarque :**

- les variables DEP, DEC, DEPTT, DETI, DEES, DERE, DEI et DEEX prennent des valeurs numériques.
- Dans le cas d'une évaluation réalisée par l'expert, ces valeurs sont prises entre 1 et 5 (elles représentent les niveaux de dommages de chaque élément).
- Pour le cas d'une évaluation réalisée par le propriétaire, ces valeurs sont prises entre 1 et 3 (elles représentent les classes de dommages pour chaque élément, c'est-à-dire 1 pour la classe vert, 2 pour la classe orange et 3 pour la classe rouge).

- **La base de règles :**

Elle regroupe les connaissances opératoires utilisées par l'expert sur lesquelles se base le moteur d'inférence pour raisonner.

Dans notre conception, nous sommes basés sur un modèle mathématique qui permet de nous calculer le dommage global de la construction.

Ce modèle est obtenu à base des plans d'expériences, dont on a utilisé 8000 fiches d'évaluation des dommages, il est constitué de trois expressions :

$$DER = a * DEP + b * DEC + c * DEPTT + d * DETI + e * DEP * DEC + f * DEP * DEPTT + g * DEC * DEPTT + h * DEP * DETI + i * DEC * DETI + j * DEPTT * DETI + k * DEP * DEC * DEPTT + l * DEP * DEC * DETI + m * DEP * DEPTT * DETI + n * DEC * DEPTT * DETI.$$

$$DES = o * DEES + p * DERE + q * DEI + r * DEEX + s * DEES * DERE + t * DERE * DEI + u * DERE * DEEX + v * DEES * DEI + w * DEES * DEEX + x * DEI * DEEX + y * DEES * DERE * DEI + z * DEES * DERE * DEEX + aa * DEES * DEI * DEEX + ab * DERE * DEI * DEEX.$$

$$DG = ac * DER + ad * DES + ae * DER * DES.$$

**Remarque :**

- Les lettres écrites en minuscule désignent les valeurs des différents coefficients, on ne les a pas mentionné pour des trucs de confidentialité.
- Les lettres écrites en majuscule désignent le nom des variables (facteurs) , elles représentent les niveaux de dommages pour une évaluation en tant qu'expert (ie. Les classes de dommages pour une évaluation en tant que propriétaire) comme suit :
  - DER : dommage éléments résistants ;
  - DEP : dommage élément porteur ;
  - DEC : dommage élément de contreventement ;
  - DEPTT : dommage élément plancher-toiture terrasse ;
  - DETI : dommage élément toiture inclinée ;
  - DES : dommage éléments secondaires ;
  - DEES : dommage élément escalier ;
  - DERE : dommage élément remplissage extérieur ;
  - DEI : dommage élément intérieur ;
  - DEEX : dommage d'autre éléments extérieurs ;
  - DG : dommage global.

Rappelons que les plans d'expériences sont des méthodes permettant de structurer le recueil d'informations pour évaluer les liens entre une variable à expliquer Y, appelée réponse, et une ou plusieurs variables explicatives X appelées facteurs (ou encore variable de prédiction selon la norme ISO 3534-3). Ces liens nous permettant par la suite d'identifier une relation de cause à effet ou d'identifier un optimum pour concevoir un meilleur produit.

Et pour cela nous avons écrit notre modèle sous forme de règles de production, donc notre base de règles est constituée de trois règles qui nous calcule respectivement, le niveau de dommages de la partie résistante, secondaire puis le niveau de dommages global de la construction comme suit :

- der(DER):- dep(DEP), dec(DEC), deptt(DEPTT), deti(DETI),
  - A is a \* DEP, B is b \* DEC,
  - C is c \* DEPTT, D is d \* DETI,
  - E is e \* DEP \* DEC,
  - F is f \* DEP \* DEPTT,
  - G is g \* DEC \* DEPTT,

H is h \* DEP \* DETI,  
 I is i \* DEC \* DETI,  
 J is j \* DEPTT \* DETI,  
 K is k \* DEP \* DEC \* DEPTT,  
 L is l \* DEP \* DEC \* DETI,  
 M is m \* DEP \* DEPTT \* DETI,  
 N is n \* DEC \* DEPTT \* DETI,  
 DER is A + B + C + D + E + F + G + H + I + J + K + L + M + N.

- des (DES):- dees(DEES), dere(DERE), dei(DEI), deex(DEEX),

A is o \* DEES, B is p \* DERE,  
 C is q \* DEI,  
 D is r \* DEEX,  
 E is s \* DEES \* DERE,  
 F is t \* DERE \* DEI,  
 G is u \* DERE \* DEEX,  
 H is v \* DEES \* DEI,  
 I is w \* DEES \* DEEX,  
 J is x \* DEI \* DEEX,  
 K is y \* DEES \* DERE \* DEI,  
 L is z \* DEES \* DERE \* DEEX,  
 M is a1 \* DEES \* DEI \* DEEX,  
 N is a2 \* DERE \* DEI \* DEEX,  
 DES is A + B + C + D + E + F + G + H + I + J + K + L + M + N.

- dg (der(DER), des (DES), DG):- der(DER), des (DES),

A is a3 \* DER,  
 B is a4 \* DES,  
 C is a5 \* DER \* DES,  
 DG is A + B + C + 1.

- **Le moteur d'inférence :**

Dans notre cas le moteur d'inférence existe déjà c'est-à-dire que nous nous occupons de mettre en place une base de connaissances sur un générateur de systèmes experts.

### II.2.B. La conception structurelle :

Après avoir conçu notre base de connaissances, nous avons choisi la conception orientée objet pour concevoir le coté structurel de notre système car elle possède de nombreux avantages : modularité, proche du langage de programmation orientée objet, conception orientée données applicable aux systèmes experts.

Ainsi nous nous basons dans notre démarche de conception, sur le langage de modélisation graphique UML.

UML, Unified Modeling Language, est un nouveau concept de modélisation objet. Son développement a débuté en 1994 par Rational Software Corporation afin d'unifier trois méthodes à objets : OOD (Object Oriented Design), OMT (Object Modeling Technique) et OOSE (Object Oriented Software Engineering). UML n'est pas entièrement nouveau, il résulte de l'évolution de ces dernières.

Nous présentons l'essentiel de l'UML dans l'annexe de ce mémoire.

- **Identification des acteurs :**

- **Définition d'un acteur : [10]**

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

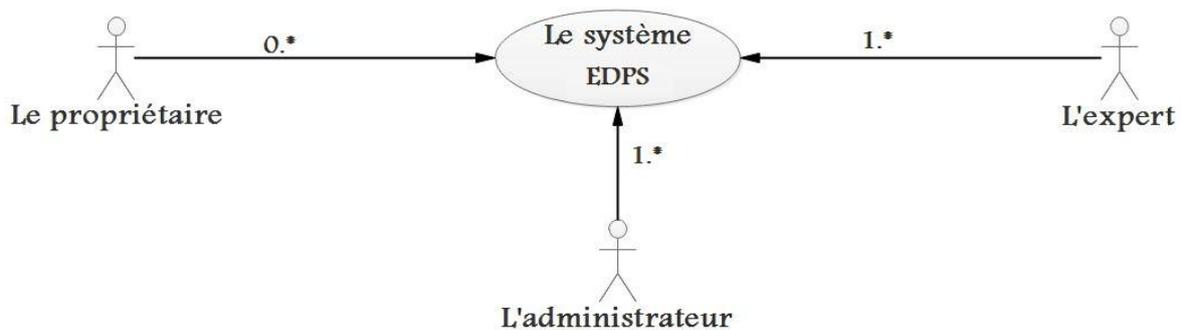
Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

- **Les acteurs de notre système :**

Notre système est utilisé par trois acteurs cités dans la section (I.3) :

- Le propriétaire ;
- L'expert ;
- L'administrateur.

Pour avoir une vision globale des interactions entre ces acteurs et le système, nous utilisons le diagramme de contexte qui est un modèle conceptuel de flux, qui est présenté dans la figure suivante :



**Figure 20: Le diagramme de contexte.**

- **Identification des cas d'utilisation : [10]**

Un cas d'utilisation correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur. Un cas d'utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou parties prenantes du système.

Le tableau suivant présente les cas d'utilisation de chaque acteur de notre système :

Acteur	Tâches
Administrateur	<p>T1 : S'authentifier en tant qu'administrateur.</p> <p>T2 : gérer les experts :  <ul style="list-style-type: none"> <li>- Ajouter un expert ;</li> <li>- Supprimer un expert ;</li> <li>- Modifier un expert.</li> </ul> </p> <p>T3 : consulter une évaluation.</p> <p>T4 : mettre à jour la base de règles.</p> <p>T5 : consulter le module d'aide.</p>
Expert	<p>T6 : S'authentifier en tant qu'expert.</p> <p>T7 : évaluer une construction en tant qu'expert.</p> <p>T8 : consulter une évaluation (identique que T3).</p> <p>T9 : consulter le module d'aide (identique que T5).</p>
Propriétaire	<p>T10 : évaluer une construction en tant que propriétaire.</p> <p>T11 : consulter une évaluation (identique que T3 et T8).</p> <p>T12 : consulter le module d'aide (identique que T5 et T9).</p>

**Tableau III.2 : Identification des cas d'utilisation.**

➤ **Diagramme de cas d'utilisation : [10]**

Tout système peut être décrit par un certain nombre de cas d'utilisation correspondant aux besoins exprimés par l'ensemble des utilisateurs. À chaque utilisateur, vu comme acteur, correspondra un certain nombre de cas d'utilisation du système. L'ensemble de ces cas d'utilisations sont représentés sous forme d'un diagramme appelé diagramme de cas d'utilisation.

La figure suivante représente le diagramme de cas d'utilisation de notre système :

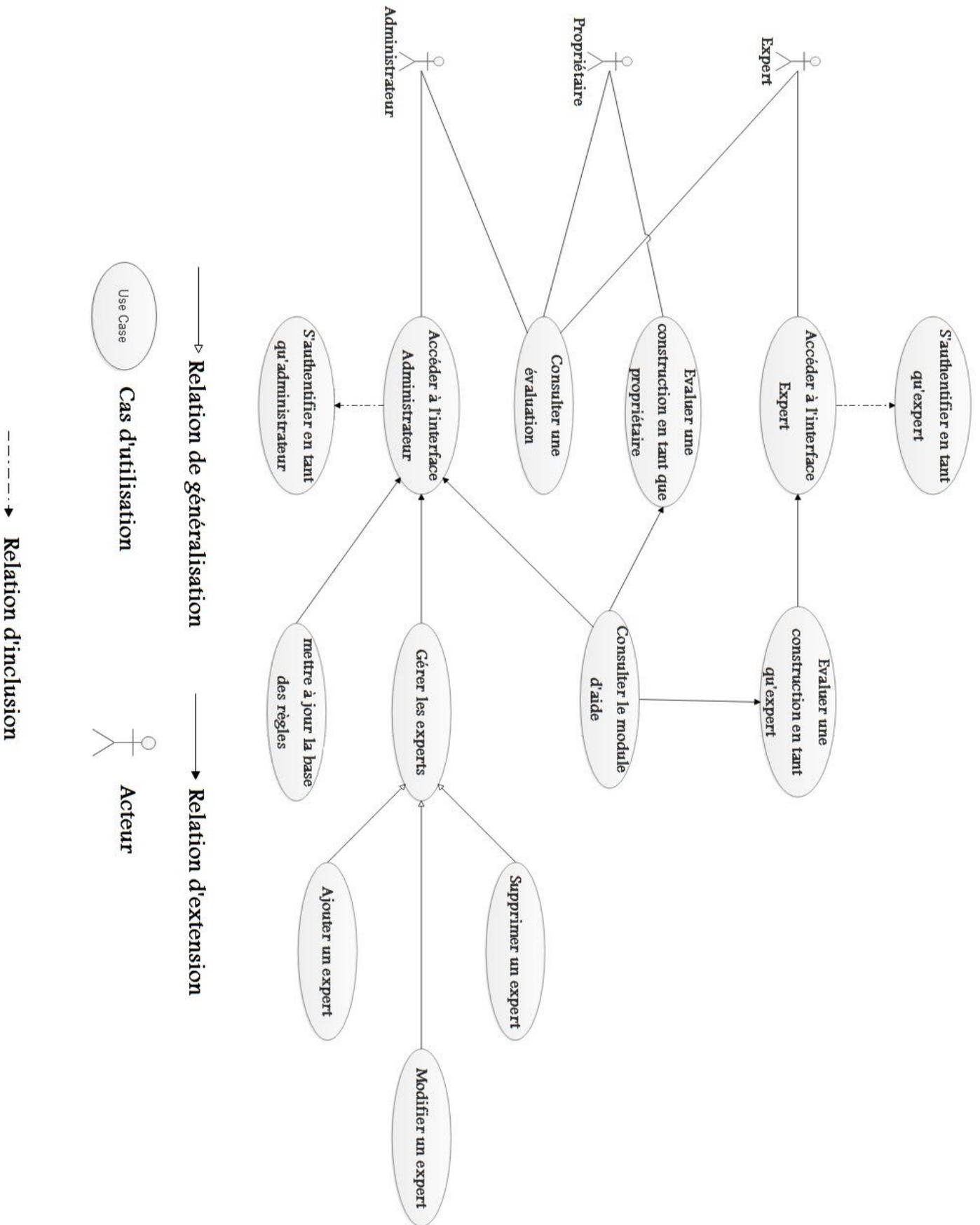


Figure 21: Diagramme de cas d'utilisation global du système EDPS.

Récapitulatif des scénarios par tâches de l'administrateur :

Tâches	Scénarios
<p><b>T1</b> - S'authentifier en tant qu'administrateur.</p>	<p><b>S1</b> - Pour accéder à son espace, l'administrateur devra d'abord s'authentifier ;  <b>S2</b> - L'administrateur demande son interface en cliquant sur le bouton « Administrateur » ;  <b>S3</b> - Le système lui retourne le formulaire d'authentification ;  <b>S4</b> - L'administrateur remplit le formulaire puis le valide en cliquant sur le bouton « Login » ;  <b>S5</b> - Le système vérifie la validité des informations reçues et retourne l'espace administrateur, sinon il retourne un message d'erreur.</p>
<p><b>T2</b> - Gérer les experts.</p>	<p><b>S1</b> - L'administrateur demande la gestion des experts en cliquant sur le bouton « gestion des experts » ;  <b>S2</b> - Le système retourne une interface contenant la listes des experts ;  <b>S3</b> - Si l'administrateur demande à modifier les informations concernant les experts en cliquant sur le bouton « modifier un expert » ;  <b>S6</b> - Le système lui affiche le formulaire de modification ;  <b>S7</b> - L'administrateur remplit les champs souhaitant être modifier puis le valide en cliquant sur le bouton « modifier » ;  <b>S8</b> - Si l'administrateur demande à supprimer un expert en cliquant sur le bouton « supprimer » ;  <b>S9</b> - Le système affiche un message de confirmation de la suppression ;  <b>S10</b> - Si L'administrateur demande l'ajout d'un expert en cliquant sur le bouton « ajouter un expert » ;  <b>S11</b> - Le système lui affiche un formulaire d'ajout d'un expert ;  <b>S12</b> -L'administrateur rempli les informations concernant l'expert à ajouter et clique sur le bouton « ajouter » ;</p>

	<p><b>S13</b> - Le système ajoute l'expert si les informations sont valides, sinon il lui affiche un message d'erreur;</p>
<p><b>T3</b> - Consulter une évaluation ;</p>	<p><b>S1</b> - L'administrateur demande la consultation d'une évaluation en cliquant sur le bouton « recherche » ;</p> <p><b>S2</b> - Le système retourne un formulaire de recherche des constructions selon plusieurs critères ;</p> <p><b>S3</b> - L'administrateur remplit le formulaire et clique sur le bouton « rechercher ».</p> <p><b>S4</b> - Le système affiche une listes des constructions ayant rempli les critères mentionnés dans le formulaire, il affiche aussi le niveau de dommage global de ces constructions.</p>
<p><b>T4</b> - Mettre à jour la base de règles ;</p>	<p><b>S1</b> - L'administrateur demande de mettre à jour la base de règles en cliquant sur le bouton « MAJ BR » ;</p> <p><b>S2</b> - Le système lui retourne l'interface de mise à jour de la base de règles et affiche les toutes les règles existantes ;</p> <p><b>S3</b> - L'administrateur met à jour la base de règles puis clique sur le bouton « valider » ;</p> <p><b>S4</b> - Le système vérifie l'inexistence de redondance et de contradiction dans la base de règles puis valide la mise à jour, un message d'erreur sera affiché dans le cas contraire ;</p>
<p><b>T5</b> - Consulter le module d'aide ;</p>	<p><b>S1</b> - L'administrateur demande le module d'aide en faisant les tâches T2, T3 ou T4 en cliquant sur le bouton « aide » ;</p> <p><b>S2</b> - Le système affiche une interface d'aide expliquant la tâche à faire.</p>

**Tableau III.3 : Récapitulatif des scénarios par tâches de l'administrateur.**

Récapitulatif des scénarios par tâches de l'expert :

Tâches	Scénarios
T6 - S'authentifier en tant qu'expert.	<p><b>S1</b> - Pour accéder à son espace l'expert devra d'abord s'authentifier.</p> <p><b>S2</b> - L'expert demande son interface en cliquant sur le bouton « Expert » ;</p> <p><b>S3</b> - Le système affiche le formulaire d'authentification;</p> <p><b>S4</b> - L'expert remplit le formulaire puis le valide en cliquant sur le bouton «Login»;</p> <p><b>S5</b> - Le système vérifie la validité des informations reçues et retourne l'espace expert, sinon il retourne un message d'erreur.</p>
T7 - Évaluer une construction en tant qu'expert.	<p><b>S1</b> - Une fois accédé à son espace, l'expert demande d'évaluer une construction en cliquant sur le bouton «nouvelle évaluation» ;</p> <p><b>S2</b> - Le système lui retourne les différents formulaires concernant l'évaluation (Renseignements généraux, Problème autour de la construction, Fondation et infrastructure, éléments résistants, éléments secondaires) ;</p> <p><b>S3</b> - Une fois l'expert remplit ces formulaires, le système lui retourne le dommage global de la construction.</p>
T8 - Consulter une évaluation	Identique que la tâche T3.
T9 - Consulter le module d'aide.	<p><b>S1</b> - L'expert demande le module d'aide en faisant les tâches T7 et T8 en cliquant sur le bouton « aide » ;</p> <p><b>S2</b> - Le système affiche une interface d'aide expliquant la tâche à faire.</p>

**Tableau III.4 : Récapitulatif des scénarios par tâches de l'expert.**

Récapitulatif des scénarios par tâches du propriétaire :

Tâches	Scénarios
T10 - Évaluer une construction en tant que propriétaire.	<p><b>S1</b> - Une fois lancé le système EDPS, le propriétaire demande d'évaluer une construction en cliquant sur le bouton «évaluer ma maison» ;</p> <p><b>S2</b> - Le système lui retourne les différents formulaires concernant l'évaluation (Renseignements généraux, Problèmes de sol, renseignements sur les constructions adjacentes, les murs extérieurs, les murs intérieurs, les escaliers, les poteaux et les poutres dans le cas d'une construction en poteaux/poutres, plancher-toiture terrasse/toiture inclinée) ;</p> <p><b>S3</b> - Une fois le propriétaire remplit ces formulaires, le système lui retourne le dommage global de la construction.</p>
T11 - Consulter une évaluation.	Identique que la tâche T3 et T8.
T12 - Consulter le module d'aide.	<p><b>S1</b> - Le propriétaire demande le module d'aide en faisant les tâches T10 et T11 en cliquant sur le bouton « aide » ;</p> <p><b>S2</b> - Le système affiche une interface d'aide expliquant la tâche à faire.</p>

**Tableau III.5 : Récapitulatif des scénarios par tâches du propriétaire.**

- **Diagrammes de séquences :**

- **Définition : [15]**

Le diagramme de séquence représente des échanges de messages entre objets. Il permet de représenter un processus de façon simplifiée, en se centrant sur les échanges entre les acteurs ou entre acteurs et le système d'information.

Dans la description des cas d'utilisation nous avons pu identifier les scénarios, dans ce qui suit nous allons traduire quelques-uns en diagrammes de séquences :

- **Diagramme de séquence de cas d'utilisation « S'authentifier en tant qu'administrateur » :**

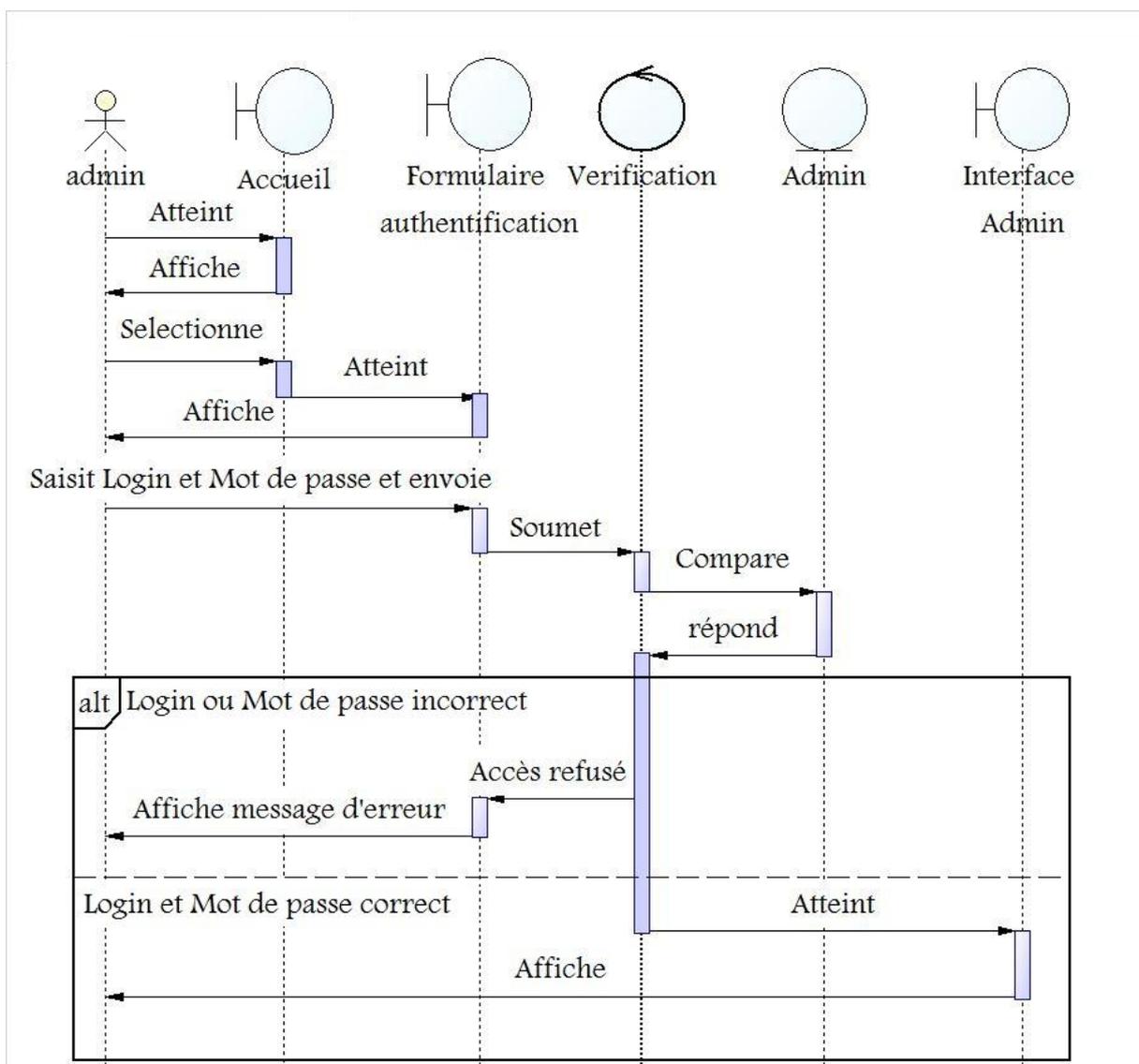
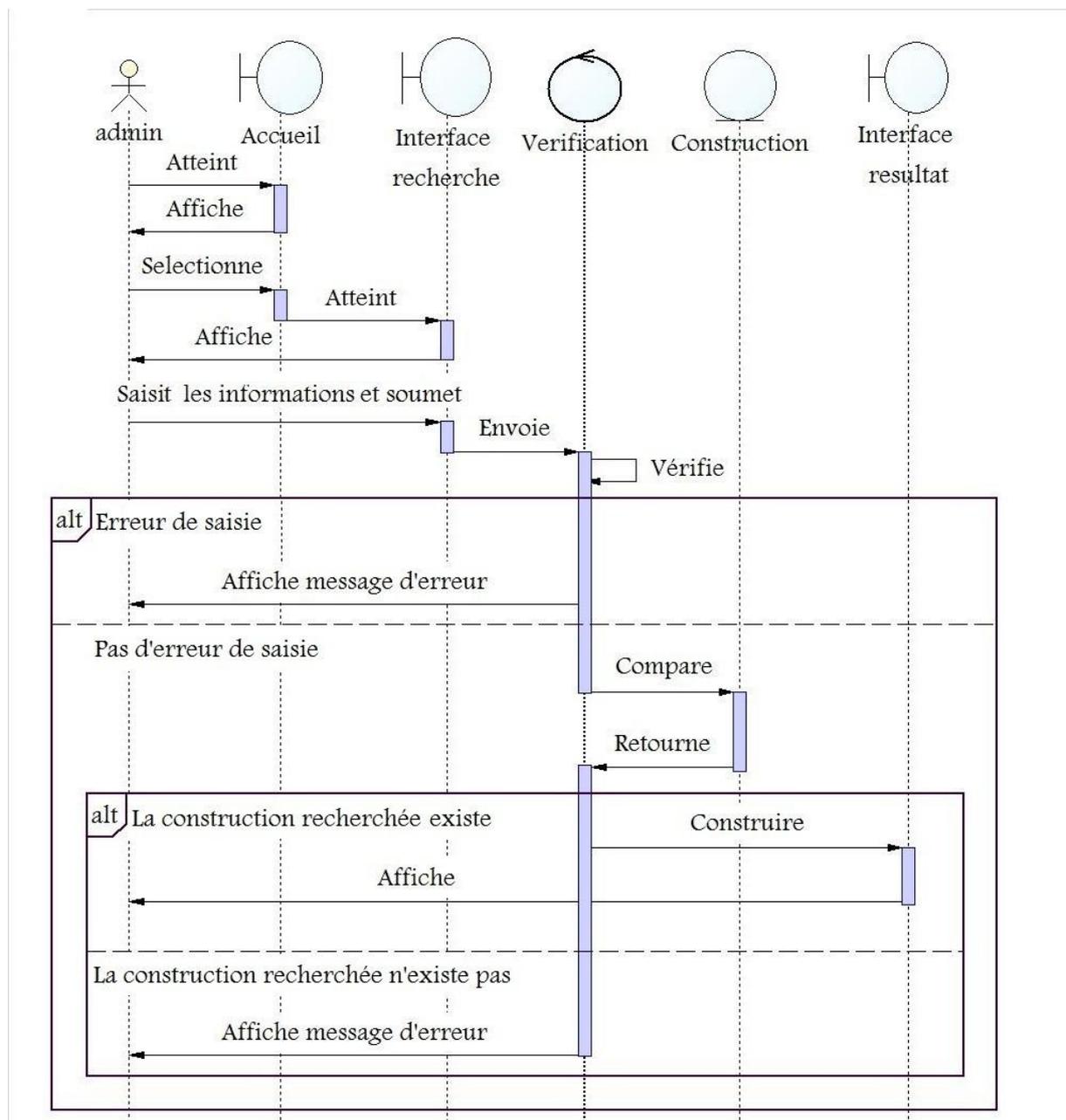


Figure 22: Diagramme de séquence de cas d'utilisation « S'authentifier en tant qu'administrateur ».

**Scénario :**

- L'administrateur accède à l'interface d'accueil en exécutant le système EDPS puis demande son espace en cliquant sur le bouton « Admin » ;
- Le système lui retourne un formulaire d'authentification ;
- L'administrateur remplit le formulaire puis le valide en cliquant sur le bouton « login »
- Le système vérifie la validité des informations reçues et redirige vers l'espace administrateur, sinon il retourne un message d'erreur.

➤ **Diagramme de séquence de cas d'utilisation « consulter une évaluation » :**



**Figure 23: Diagramme de séquence de cas d'utilisation « consulter une évaluation ».**

### **Scénario :**

- L'administrateur accède à l'interface d'accueil en exécutant le système EDPS puis demande l'interface recherche en cliquant sur le bouton « recherche » ;
- Le système lui retourne l'interface recherche contenant un formulaire ;
- L'administrateur remplit le formulaire et valide l'opération ;
- Le système vérifie les données saisies et retourne l'interface résultat si cette construction existe, ce dernier affiche un message d'erreur dans le cas contraire.

➤ Diagramme de séquence de cas d'utilisation « Ajouter un expert » :

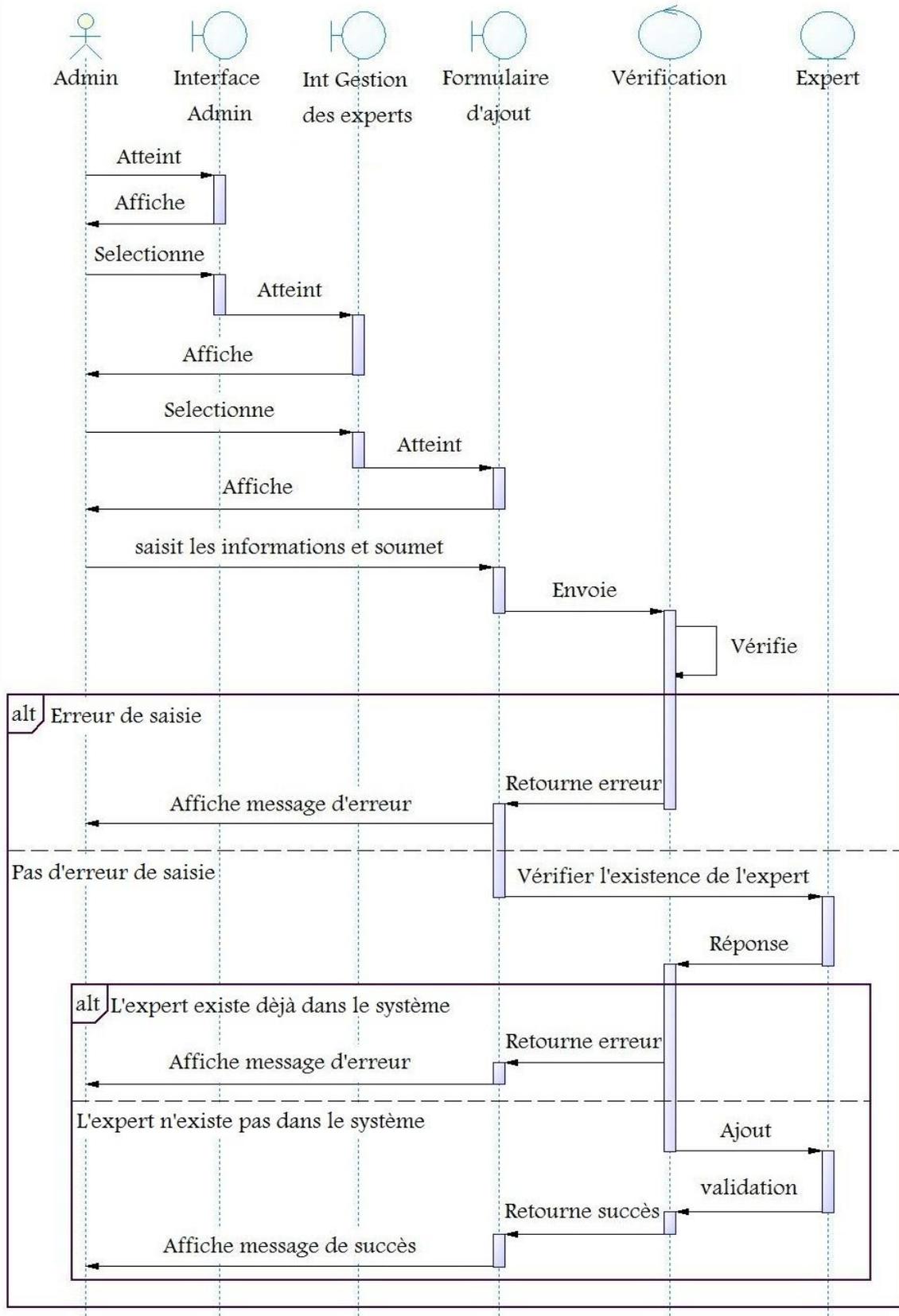


Figure 24: Diagramme de séquence de cas d'utilisation « ajouter un expert ».

**Scénario :**

- Après avoir authentifié, l'administrateur demande l'interface « Gestion des Experts » en cliquant sur le bouton « Gestion Experts » ;
- Le système lui retourne l'interface « Gestion des Experts » ;
- L'administrateur demande le formulaire d'ajout en cliquant sur le bouton « Ajouter Expert » ;
- Le système lui retourne le formulaire ;
- L'administrateur remplit le formulaire et le valide l'opération ;
- Le système vérifie les données saisies puis vérifie l'inexistence de cet expert dans la base de données, si c'est le cas, le système enregistre le nouvel expert dans la base de données et affiche un message de succès, sinon il affiche un message d'erreur.

- **Diagramme de séquence de cas d'utilisation « évaluer une construction en tant qu'expert » :**

**Remarque :**

Dans ce diagramme nous avons utilisé quelques abréviations tels que :

- RGC : Renseignements généraux sur la construction ;
- Sol-C.Adja : Sol et Constructions Adjacentes ;
- Fond-Infra : Fondations et Infrastructures ;
- Struct-Resist : Structure Résistante ;
- Struct-Second : Structure Secondaire.

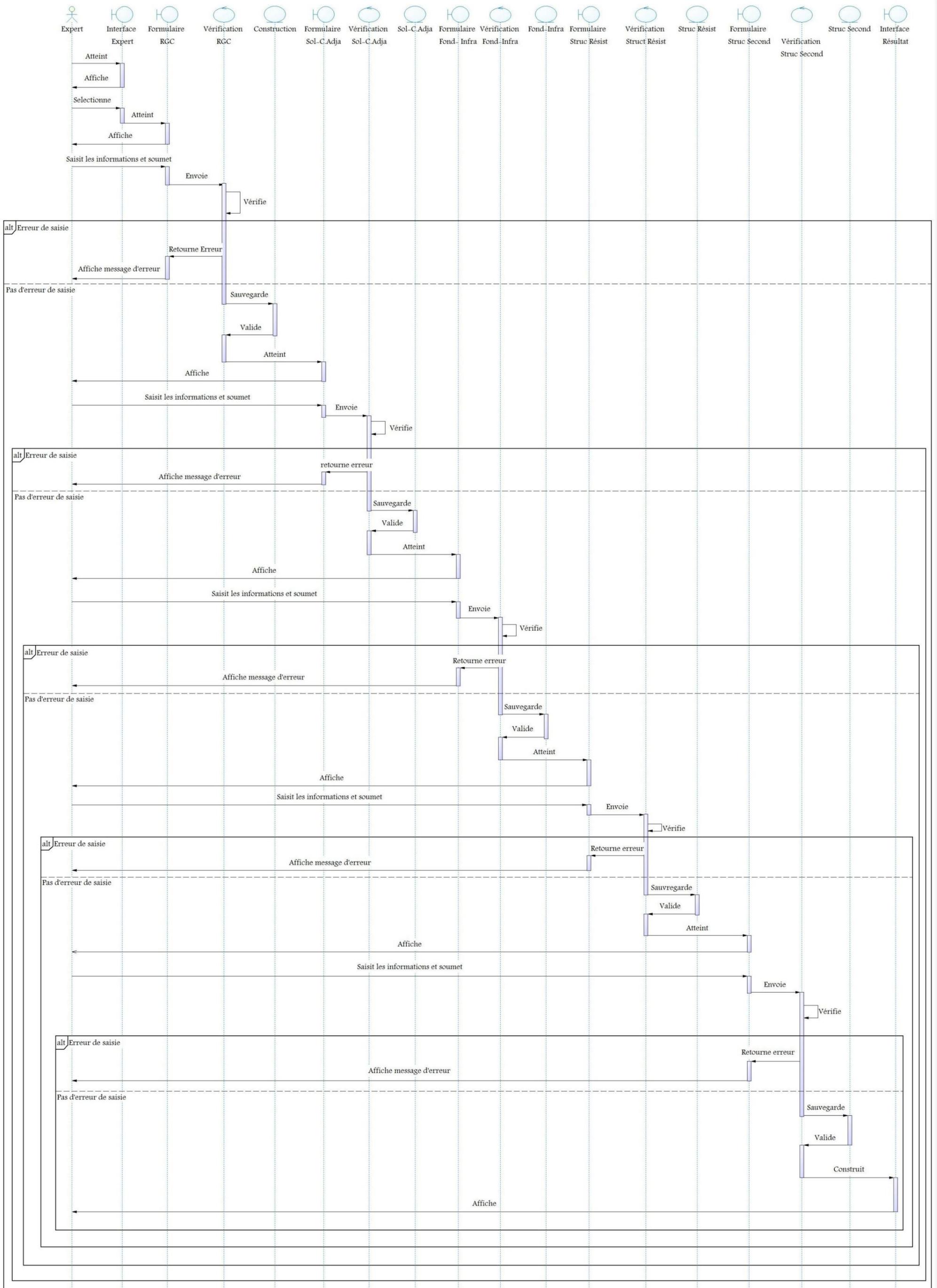


Figure 25: Diagramme de séquence de cas d'utilisation « évaluer une construction en tant qu'expert ».

**Scénario :**

- Une fois accédé à son espace, l'expert demande l'évaluation d'une nouvelle construction en cliquant sur le bouton « nouvelle évaluation » ;
- Le système lui retourne l'interface RGC (renseignements généraux sur la construction), cette dernière contient un formulaire;
- L'expert remplit le formulaire puis le valide ;
- Le système vérifie les données saisies, puis les enregistre dans la BDD et affiche le formulaire du sol et constructions adjacentes, un message d'erreur sera affiché dans le cas contraire ;
- L'expert remplit le formulaire puis le valide ;
- Le système vérifie les données saisies, puis les enregistre dans la BDD et affiche le formulaire des fondations et infrastructure, un message d'erreur sera affiché dans le cas contraire ;
- Le système vérifie les données saisies, puis les enregistre dans la BDD et affiche le formulaire de la structure résistante, un message d'erreur sera affiché dans le cas contraire ;
- L'expert remplit le formulaire puis le valide ;
- Le système vérifie les données saisies, puis les enregistre dans la BDD et affiche le formulaire de la structure secondaire, un message d'erreur sera affiché dans le cas contraire ;
- L'expert remplit le formulaire puis le valide ;
- Le système vérifie les données saisies, puis les enregistre dans la BDD et affiche l'interface résultat (cette interface contient le résultat de l'évaluation c'est-à-dire son niveau de dommages global), un message d'erreur sera affiché dans le cas contraire ;

- **Le diagramme de classe : [15]**

Les diagrammes de classe sont sans doute les diagrammes les plus utilisés d'UML. Ils décrivent les types des objets qui composent un système et les différents types de relations statiques qui existent entre eux. Les diagrammes de classe font abstraction du comportement du système.

Pour le cas de notre application, le diagramme de classe est illustré dans la figure suivante :

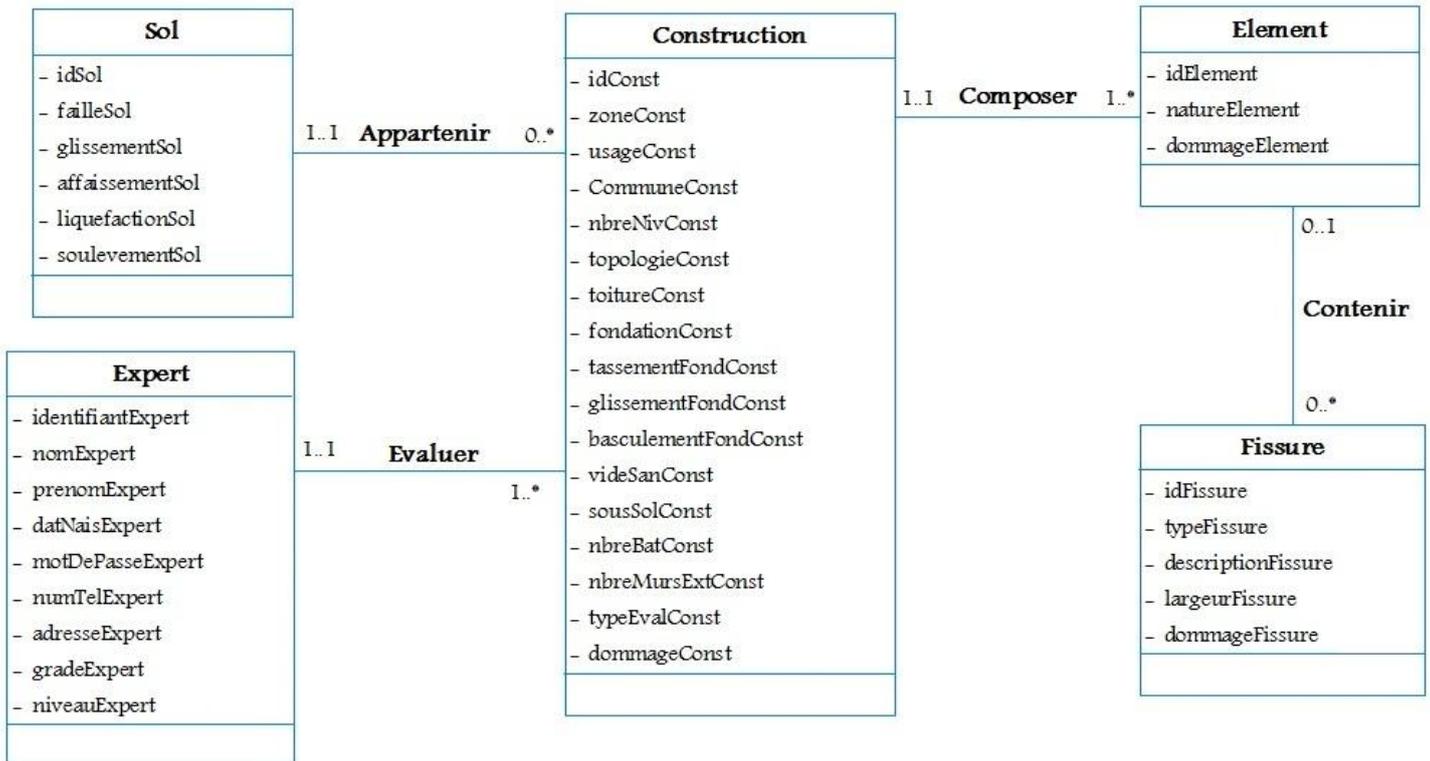


Figure 26: Diagramme de classe du système EDPS.

- **Le modèle relationnel :**

En appliquant les règles de transformation d'un diagramme de classe vers un modèle relationnel, nous avons aboutit au schéma relationnel suivant :

- **Construction**(idConst, zoneConst, communeConst, nbreNivConst, topologieConst, toitureConst, fondationConst, tassementFondConst, glissementFondConst, basculementFonConst, videSanConst, sousSolConst, nbreMursExtConst, typeEvalConst, dommageConst, identifiantExpert\*) ;
- **Element**(idElement, natureElement, dommageElement, idConst\*) ;
- **Expert**(identifiantExpert, prenomExpert, dateNaisExpert, numTelExpert, motDePasseExpert, adresseExpert, gradeExpert, niveauExpert) ;
- **Fissure**(idFissure, typeFissure, descriptionFissure, largeurFissure, dommageFissure, idElement\*) ;
- **Sol**(idSol, failleSol, affaissementSol, glissementSol, liquefactionSol, soulevementSol, idConst\*) ;

**Remarque :** les clés primaires sont soulignées, et les clés étrangères sont marquées par une étoile (\*).

**III. Conclusion :**

Dans cette partie nous avons fait l'étude du futur système (analyse et conception), l'implémentation étant la partie qui suit celle de la conception détaillée, elle nous permettra de réaliser le futur système. En effet dans la partie suivante, on fera la description de la dernière phase du projet qui est l'implémentation.

*CHAPITRE* **IV**

---

***RÉALISATION***

## Introduction :

Après avoir présenté les détails de la conception du système EDPS dans le chapitre précédent, nous passons maintenant à la phase de réalisation qui consiste à implémenter les différentes fonctionnalités du système.

Ce chapitre sera partitionné en trois parties, la première partie, présentera les différents outils de développement utilisés dans notre réalisation, la 2<sup>ème</sup> partie présentera les multiples fonctionnalités qu'offre le système à travers différentes interfaces, tandis que la 3<sup>ème</sup> partie sera consacrée aux tests de notre système.

## I. Description des outils de travail :

### I.1. Aspect matériel :

Notre système « EDPS » a été développé sous les contraintes matérielles suivantes :

- Micro-ordinateur portable de marque TOSHIBA ;
- Taille de disque dur : 500 Go ;
- Capacité de mémoire vive : 04 Go ;
- Processeur : Intel ® Core™ i3-3120M CPU @ 2.5 GHz 2.5 GHz.

### I.2. Aspect logiciel :

#### I.2.A. Le système d'exploitation :

Notre environnement de travail est doté du système **Windows 7 Version 64 bits**. Cet environnement se caractérise par un interfaçage facile à utiliser, et un développement d'applications simple et rapide.

#### I.2.B. Environnement de développement :

Dans le système EDPS, le raisonnement du MI se fait sur les faits et règles de la BC. Pour représenter la BC, nous avons opté pour le formalisme logique à travers les prédicats Prolog.

Le raisonnement est alors assuré par le moteur d'inférence Prolog. La base de travail est alimentée interactivement via des interfaces Java. L'interfaçage entre Java et Prolog est assuré par une bibliothèque spécifique : la JPL (Java Prolog Link).

Afin de sauvegarder les informations et les renseignements concernant les différentes constructions et les dommages qu'elles ont subies, nous avons mis en place une base de données relationnelle implémentée dans le système de gestion des bases de données relationnelles (SGBDR) MySQL, ce SGBDR est inclus dans la plateforme WAMP. La connectivité entre le serveur MySQL et Java (la manipulation de la BDD à partir de NetBeans) est assurée par l'API Java qui s'appelle JDBC (Java DataBase Connectivity).

L'environnement de développement de notre application est constitué donc de :

- Prolog (SWI-Prolog 7.4.1) pour la programmation de la base de connaissances, et le raisonnement logique ;

- Java sous NetBeans IDE 8.2 pour la programmation des interfaces graphiques ;
- JPL (Java) pour l'interfaçage Java/Prolog ;
- MySQL pour l'implémentation de la BDD ;
- JDBC (Java) pour la liaison Java/MySQL.

### I.3. Prolog :

#### I.3.A. Définition : [11]

Prolog « PROgrammation LOGique » est né en 1972 à l'université de Marseille conçu par Alain Colmerauer et son équipe. Prolog a hérité de la majorité des fonctionnalités du langage Lisp mais avec une syntaxe plus adorable.

Un programme Prolog consiste en la déclaration d'une série de faits et de règles de déduction. L'exécution du programme consiste à prouver un théorème. Le programme répond si le théorème peut être prouvé ou non à partir des déclarations.

Le langage d'expression des connaissances de Prolog est fondé sur le langage des prédicats du premier ordre. En tant que moteur d'inférence, Prolog intègre deux mécanismes puissants : l'unification pour réaliser le filtrage des connaissances, et le retour arrière dans l'enchaînement des cycles d'inférence.

Plusieurs implémentations de Prolog existent, nous avons opté pour la version gratuite SWI-Prolog.

#### I.3.B. SWI-Prolog : [17]

SWI-Prolog est une implémentation complète et portable du langage de programmation Prolog développé par Jan Wielemaker à l'université d'Amsterdam, qui est distribué gratuitement (licence GPL). SWI-Prolog vise à être une implémentation robuste et évolutive prenant en charge un large éventail d'applications. En particulier, elle est livrée avec une large gamme de bibliothèques d'interface, fournissant interfaces à d'autres langages, d'autres bases de données, des graphiques et réseaux. Elle fournit un support étendu pour la gestion des documents de type HTML/SGML/XML et RDF. Le système est particulièrement adapté aux applications serveur grâce à un support robuste pour les bibliothèques multithreading et serveurs HTTP.

Cette version est caractérisée par :

- Une syntaxe simple ;
- Il offre divers possibilités de communication avec d'autres langages comme le C, le C++ et Java ;
- Un débogueur complet ;
- Une documentation complète et gratuite.

Pour pouvoir fonctionner, un programme SWI-PROLOG<sup>®</sup> doit obéir à certaines conventions :

- Tous les faits et règles doivent se terminer par un «.».

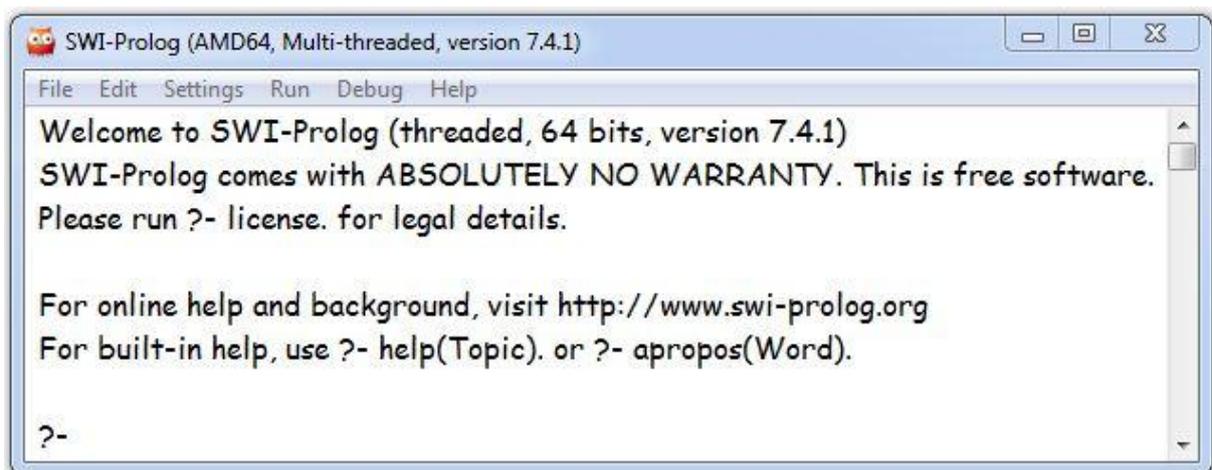
- Les variables utilisées dans les faits ou les règles commencent toujours par une majuscule ou « \_ ».

**Remarque 1:** Mieux vaut faire commencer les variables classiques par des majuscules. En effet, cela permet de différencier sans ambiguïté dans le mode débogage les variables du programme, des variables internes générées dynamiquement par PROLOG qui commencent par « \_ ».

- A l'inverse, tout le reste commence par une minuscule.
- Il ne faut pas mettre d'espace entre le prédicat et la parenthèse ouvrante qui l'accompagne.
- Les commentaires sont mis entre « /\* » et « \*/ » ou commencent par « % » et se terminent sur la même ligne.

**Remarque 2:** Ces conventions s'appliquent également aux requêtes faites par l'utilisateur dans l'interpréteur de commande de SWI-PROLO.

Voici une capture d'écran de l'interface de SWI-Prolog :



**Figure 27: L'IDE SWI-Prolog.**

## I.4. Java :

### I.4.A. Définition : [8]

Java est un langage de programmation orienté objet développé par Sun Microsystems en 1995 (puis racheté par Oracle Corporation en avril 2009). Sa syntaxe est proche de celle du langage C++ mais avec des fonctionnalités qui rendent le langage plus simple : absence de pointeurs, gestion automatique de la mémoire centrale.

Trois plateformes d'exécution (ou éditions) Java sont définies pour des cibles distinctes selon les besoins des applications à développer :

- Java Standard Edition (J2SE / Java SE) : environnement d'exécution et ensemble complet d'API pour des applications de type desktop. Cette plate-forme sert de base en tout ou partie aux autres plateformes ;

- Java Enterprise Edition (J2EE / Java EE) : environnement d'exécution reposant intégralement sur Java SE pour le développement d'applications d'entreprises ;
- Java Micro Edition (J2ME / Java ME) : environnement d'exécution et API pour le développement d'applications sur appareils mobiles et embarqués dont les capacités ne permettent pas la mise en œuvre de Java SE.

La séparation en trois plateformes permet au développeur de mieux cibler l'environnement d'exécution et de faire évoluer les plateformes de façon plus indépendante.

#### **I.4.B. Historique : [6]**

On peut faire remonter la naissance de Java à 1991. À cette époque, des ingénieurs de chez Sun ont cherché à concevoir un langage applicable à de petits appareils électriques (on parle de *code embarqué*). Pour ce faire, ils se sont fondés sur une syntaxe très proche de celle de C++, en reprenant le concept de machine virtuelle déjà exploité auparavant par le Pascal UCSD. L'idée consistait à traduire d'abord un programme source, non pas directement en langage machine, mais dans un pseudo langage universel, disposant des fonctionnalités communes à toutes les machines. Ce code intermédiaire, dont on dit qu'il est formé de *bytecodes*, se trouve ainsi compact et portable sur n'importe quelle machine ; il suffit simplement que cette dernière dispose d'un programme approprié (on parle alors de *machine virtuelle*) permettant de l'interpréter dans le langage de la machine concernée.

En fait, ce projet de langage pour code embarqué n'a pas abouti en tant que tel. Mais ces concepts ont été repris en 1995 dans la réalisation du logiciel *HotJava*, un navigateur Web écrit par Sun en Java, et capable d'exécuter des *applets* écrits précisément en *bytecodes*.

Les autres navigateurs Web ont suivi, ce qui a contribué à l'essor du langage qui a beaucoup évolué depuis cette date, sous forme de versions successives :

- 1996 : JDK 1.0
- 1997 : JDK 1.1
- 1998 : JDK 1.2, appelé Java 2
- 2000 : JDK 1.3
- 2002 : JDK 1.4
- 2004 : JDK 1.5, appelé Java 5
- 2006 : JDK 1.6, appelé Java 6
- 2011 : JDK 1.7, appelé Java 7
- 2014 : JDK 1.8, appelé Java 8.

#### **I.4.C. Qu'est ce que le JDK ? :**

L'écriture des applets et des applications Java nécessite l'utilisation d'outils de développement tels que le kit JDK. Ce dernier comprend l'environnement JRE (Java Runtime Environment), le compilateur Java et les API Java.

Le JDK (Java Development Kit) est l'environnement minimal pour écrire des programmes Java, le JDK contient :

- javac: c'est un compilateur, c'est-à-dire qu'il transforme le code source en bytecodes ;
- java: L'interpréteur java est la machine virtuelle permettant de lire le bytecode ;
- applet viewer: l'interpréteur d'applets, il a pour but de pouvoir visualiser l'exécution d'un applet (il est aussi possible de la visualiser sur la plupart des navigateurs internet) ;
- jdb: un débogueur, c'est un outil permettant d'inspecter l'exécution du programme Java ;
- javap: un décompilateur, pour revenir du bytecode au code source ;
- javadoc: le générateur de documentation, c'est un utilitaire permettant de créer une documentation au format HTML à partir des programmes Java;
- jar: l'éditeur d'archives Java, c'est est un utilitaire permettant de compresser toutes les classes d'un même projet dans une archive (formellement, un fichier zip avec l'extension .jar) ;
- le JDK contient aussi un certain nombre de packages, chacun étant constitué par un ensemble de classes qui couvrent un même domaine et apportent de nombreuses fonctionnalités, parmi ces packages on peut citer : java.lang, java.util, java.io, java.sql, java.math, java.nio, javax.swing ... etc.

#### **I.4.D. Les caractéristiques du langage Java : [8]**

Java possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès :

- Java est interprétée : le code source est compilé en bytecode puis exécuté par un interpréteur Java, la Java Virtual Machine (JVM). Ce concept est à la base du slogan de Sun pour Java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). En effet, le bytecode, s'il ne contient pas de code spécifique à une plate-forme particulière peut être exécuté et obtenir quasiment les mêmes résultats sur toutes les machines disposant d'une JVM.
- Java est portable : il est indépendant de toute plate-forme, il n'y a pas de compilation spécifique pour chaque plate forme. Le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine. Cette indépendance est assurée au niveau du code source grâce à Unicode et au niveau du bytecode.
- Java est orienté objet : comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).

- Java est simple : le choix de ses auteurs a été d'abandonner des éléments mal compris ou mal exploités des autres langages tels que la notion de pointeurs (pour éviter les incidents en manipulant directement la mémoire), l'héritage multiple et la surcharge des opérateurs, ...
- Java est fortement typé : toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données. Si une telle conversion doit être réalisée, le développeur doit obligatoirement utiliser un cast ou une méthode statique fournie en standard pour la réaliser.
- Java assure la gestion de la mémoire : l'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au garbage collector qui restitue les zones de mémoire laissées libres suite à la destruction des objets.
- Java est économe : le pseudo code possède une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution.
- Java est multitâche : il permet l'utilisation de threads qui sont des unités d'exécutions isolées. La JVM, elle même, utilise plusieurs threads.

#### **I.4.E. Java et les bases de données : [8]**

Afin de permettre à un programmeur Java de développer des applications orientées base de données, la plateforme Java intègre une API appelée JDBC à partir de la version 1.1 du JDK.

JDBC est l'acronyme de Java DataBase Connectivity. C'est l'API standard pour permettre un accès à une base de données. Son but est de permettre de coder des accès à une base de données en laissant le code le plus indépendant de la base de données utilisée.

C'est une spécification qui définit des interfaces (package Java.sql) pour se connecter et interagir avec la base de données (exécution de requêtes ou de procédures stockées, parcours des résultats des requêtes de sélection, ...).

L'implémentation de ces spécifications est fournie par des tiers, et en particulier les fournisseurs de base de données, sous la forme de pilote (Driver).

Il existe 4 types de pilotes JDBC :

- Type 1 : pont JDBC – ODBC ;
- Type 2 : pilote qui fait appel à des fonctions natives (code non Java, le plus souvent en C ou C++) de l'API du SGBD ;
- Type 3 : pilote qui permet l'utilisation d'un serveur middleware ;
- Type 4 : pilote entièrement en Java qui utilise directement le protocole réseau du SGBD.

Les drivers se présentent souvent sous forme de fichiers jar dont le chemin doit être ajouté au classpath pour permettre au programme de l'utiliser.

Pour notre cas nous utilisons « MySQL JDBC Driver MySQL-Connector-Java-5.1.23 », qui est de type 4.

Toutes les classes de JDBC sont dans le package `java.sql`. Il faut donc l'importer dans tous les programmes devant utiliser JDBC. Il y a 4 classes importantes : `DriverManager`, `Connection`, `Statement` (et `PreparedStatement`), et `ResultSet`, chacune correspondant à une étape de l'accès aux données :

Classe	Rôle
<code>DriverManager</code>	Charger et configurer le driver de la base de données.
<code>Connection</code>	Réaliser la connexion et l'authentification à la base de données.
<code>Statement</code> (et <code>PreparedStatement</code> )	Contenir la requête SQL et la transmettre à la base de données.
<code>ResultSet</code>	Parcourir les informations retournées par la base de données dans le cas d'une sélection de données.

Chacune de ces classes dépend de l'instanciation d'un objet de la précédente classe.

#### I.4.F. Présentation de l'EDI NetBeans : [19]

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000.

En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, C++, JavaScript, XML, Groovy, PHP, HTML, ou d'autres (dont Python et Ruby) par l'ajout de plugins.

Compilé en Java, NetBeans est disponible sous Windows, Linux, Solaris, Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).

C'est un IDE moderne offrant un éditeur avec des codes couleurs et un ensemble de signes, des modèles de projets multi-langage et de différents types (application indépendante, distribuée, plugin, mobiles, ...), le refactoring, l'éditeur graphique d'interfaces et de pages web pour supporter le programmeur dans son travail. Il permet d'accéder rapidement à la documentation détaillée, de naviguer dans les sources et de faire des recherches d'usage des classes, méthodes et propriétés.

NetBeans indique à l'utilisateur les erreurs et fait des propositions pour y remédier. Un débogueur permet l'exécution pas à pas. Un suivi des ressources utilisées (CPU, Mémoire) par le logiciel développé peut être fait via un profiler. Un Framework de test unitaire tel que Junit Fiche Junit peut être utilisé.

L'EDI NetBeans fournit des outils pour construire tous les composants Java EE, ce qui inclut les Enterprise JavaBeans (EJBs), les pages Web, les Servlets, et les Services Web.

Il intègre le serveur d'application Glassfish, ce qui permet de facilement développer des EJB et de les déployer. Il intègre aussi la norme Webservice JAX-WS. Il est aisé de lier un WS avec un EJB pour faire son implémentation.

C'est pour ces raisons que nous l'avons adopté pour le développement de nos interfaces.

La figure ci-dessous illustre l'interface de NetBeans :

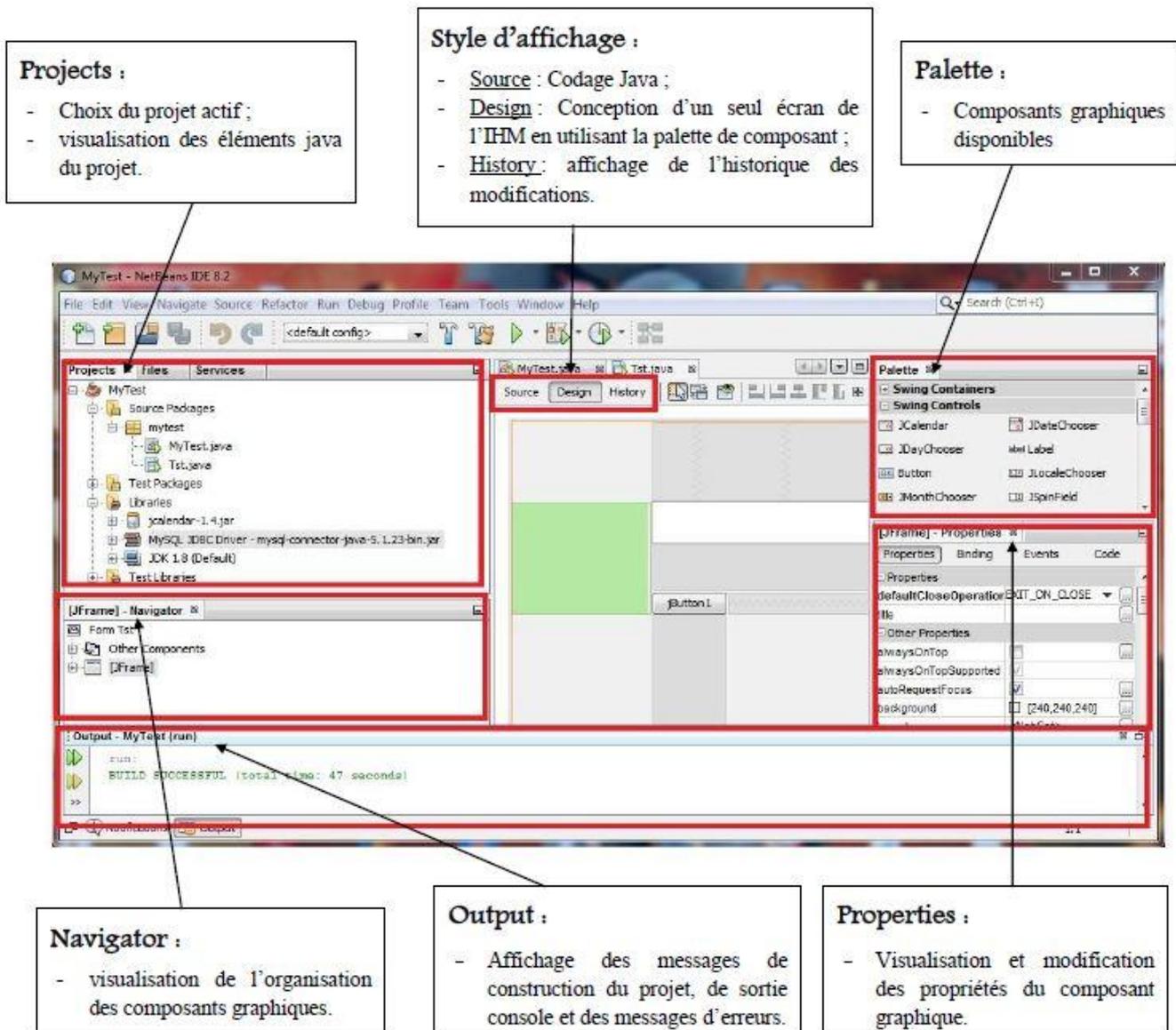


Figure 28: L'interface de l'IDE NetBeans 8.2.

### I.5. Interfaçage entre SWI-Prolog et Java :

Pour rendre SWI-Prolog complet du point de vue des interfaces graphiques et portatif, les créateurs de SWI-Prolog ont mis en place une bibliothèque appelée **JPL** qui permet à java de communiquer avec prolog et vice-versa.

#### I.5.A. Qu'est ce que la JPL : [22]

La JPL est un ensemble de classes Java et de fonctions C fournissant une interface entre Java et prolog.

La JPL emploie la JNI (Java Native Interface) pour se relier au moteur de prolog à travers son interface FLI des langages externes (Foreign Language Interface).

Elle est conçue en deux couches différentes, une interface de bas niveau en prolog FLI et une interface de haut niveau en Java (cette dernière interface est destinée aux programmeurs Java qui ne sont pas concernés par les détails de la programmation logique et le fonctionnement de la FLI).

L'architecture des classes de la JPL est présentée dans la figure suivante :

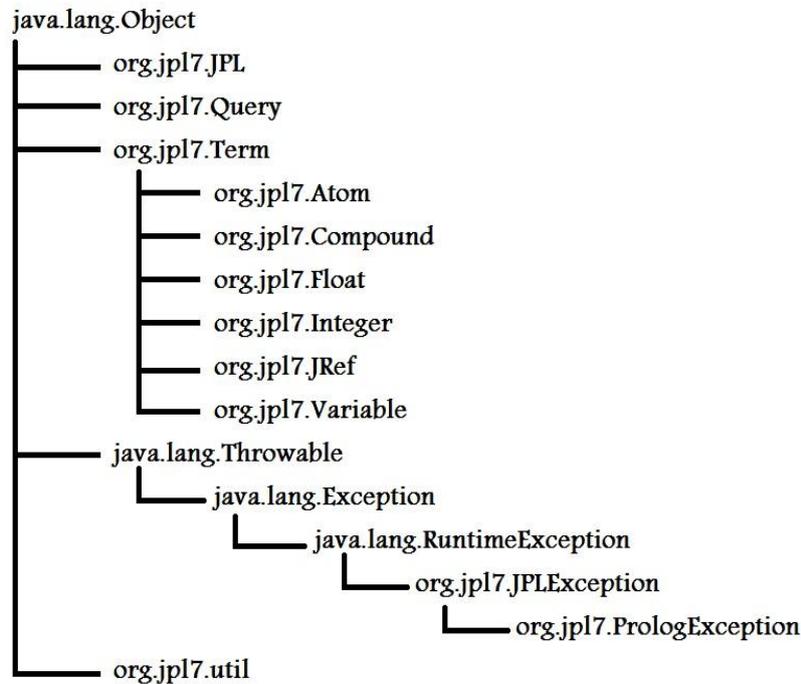


Figure 29: Architecture des classes de la JPL.

La figure suivante présente le schéma de l'API JPL:

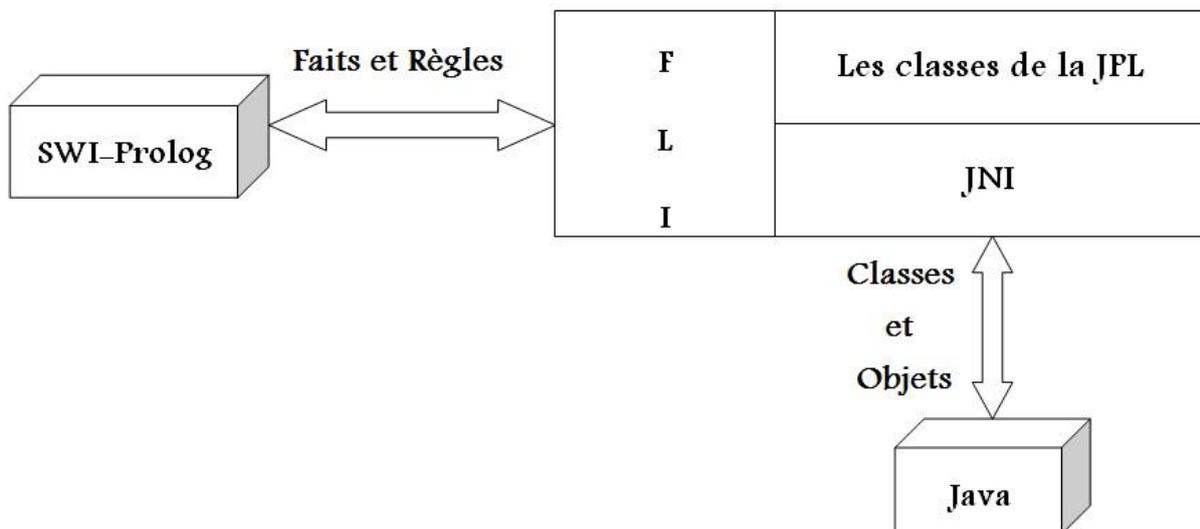


Figure 30: Schéma de la JPL.

**I.5.B. Présentation générale de quelques classes utilisées de la JPL : [22]**

- **la classe Term** : C'est la classe de base abstraite pour Compound, Atom, Variable, Integer et Float, qui comprend une syntaxe concrète orientée Java pour Prolog.

Chaque Objet instancié de cette classe représente un atome, un entier ou un flottant dans Prolog.

**Exemple :**

- `Term but = new Atom ("femme(djamila)");`

- **La classe Atom** : Chaque objet instancié de la classe atome représente un atome de Prolog avec le même nom. Un atome est construit avec un paramètre de Type String qui ne peut pas ensuite être changé.

**Exemple :**

- `Atom atm= new Atom("dommage_element_secondaire") ;`

- **La classe Variable** : Comme son nom l'indique, elle sert à déclarer des Variables.

**Exemple :**

- `Variable X = new Variable ("X") ; //Déclaration de la variable nommée X.`

- **La classe Query** : Elle représente une question ou une requête pour la base de connaissances, elle a comme arguments soit un terme ou bien un atome, ou chaîne de caractères.

**Exemple :**

- `Query q = new Query (but) ; //Dans le cas d'un terme.`
- `Query q = new Query ("femme (djamila)"); //Dans le cas d'une chaîne de caractères.`

- **La classe Float** : Float est un terme spécialisé avec un champ Double, représentant une valeur de virgule flottante ISO / IEC Prolog 64 bits. Une fois construit, sa valeur ne peut pas être modifiée.

**Exemple :**

- `Float f = new Float (3.14159265) ; // Déclaration d'un nombre en virgule flottante.`

- **La classe Integer** : Integer est un terme spécialisé représentant une valeur entière Prolog, Une fois construit, sa valeur ne peut pas être modifiée.

**Exemple :**

- `Integer nbre = new Integer (5) ; // Déclaration d'un nombre entier.`

- **La classe Util :** Cette classe fournit un ensemble de méthodes supplémentaires, par exemple : la méthode `textToTerm`, `textParamsToTerm`, `termArrayToList`, `stringArrayToList` ... Ces méthodes permettent de convertir et de basculer entre différent types.
- **La classe Compound :** c'est permet de déclarer un terme structuré, c'est-à-dire il est comprend un foncteur et un ou plusieurs arguments (termes).

**Exemple :**

- `Compound cmp = new Compound ("père_de", new Term[]{new Atom("djamila"),new Atom("ali")} );`

**Remarque:** La JPL gère un ensemble d'exceptions regroupées dans la Classe `JPLException`.

## I.6. MySQL :

### I.6.A. Définition : [23]

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

Son nom vient du prénom de la fille du co-créateur Michael Widenius, My. SQL fait référence au Structured Query Language, le langage de requête utilisé.

MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle DataBase et MySQL. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010.

Pour réaliser notre projet, nous avons utilisé la version gratuite de MySQL, cette dernière est incluse dans la plateforme WAMP.

### I.6.B. WAMP : [24]

WAMP Server est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

### I.6.C. PHPMyAdmin : [25]

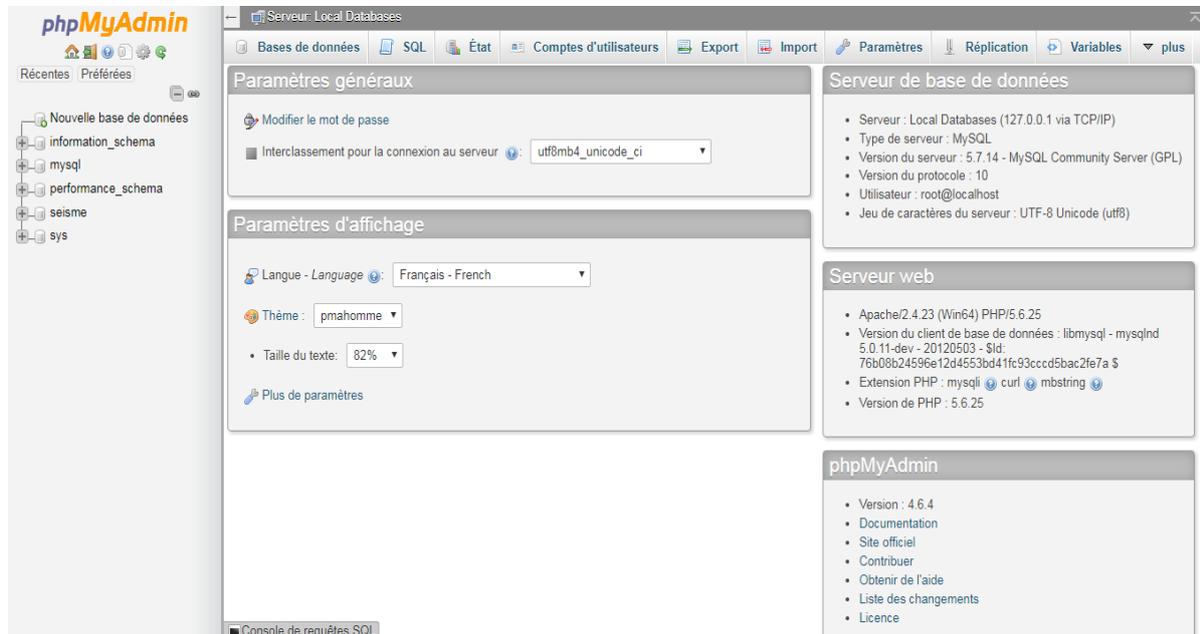
PHPMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, gratuits comme payants, le proposent ce qui évite à l'utilisateur d'avoir à l'installer.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de

sauvegarder commodément une base de données sous forme de fichier .SQL et d'y transférer ses données, même sans connaître SQL.

La figure suivante présente l'interface de PHPMYAdmin :



**Figure 31: L'interface PHPMYAdmin.**

## II. Présentation des interfaces du logiciel :

Nous allons présenter dans ce qui suit, les principales interfaces illustrant le fonctionnement de notre logiciel.

### II.1. L'interface d'accueil :

Après le lancement de notre logiciel, la fenêtre ci-dessous apparaîtra, avec une passerelle vers les autres interfaces.



Figure 32: Interface d'accueil du logiciel.

## II.2. L'interface d'authentification pour l'expert :

Quand on appuie sur le bouton « Expert », une interface d'authentification s'apparaît, demandant à l'expert de saisir son identifiant ainsi que son mot de passe.

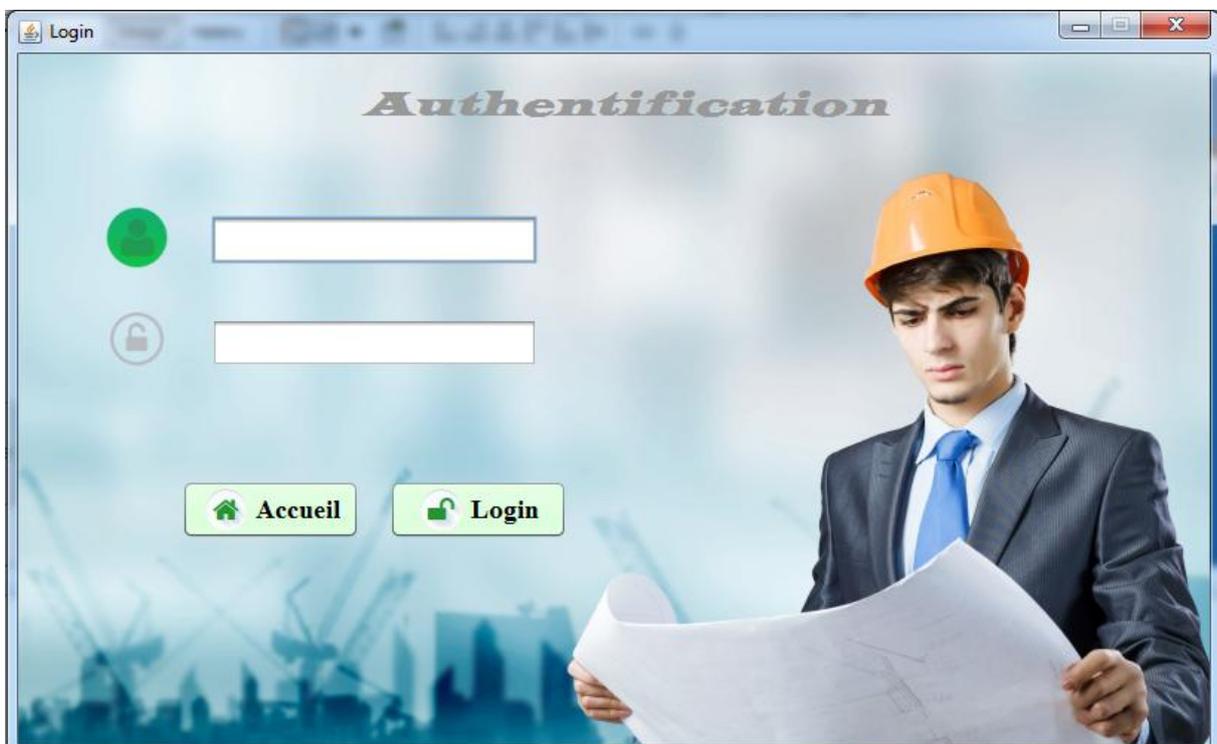


Figure 33: Interface d'authentification de l'expert.

### II.3. L'interface Expert :

Une fois authentifié, l'accès est autorisé à l'interface Expert.

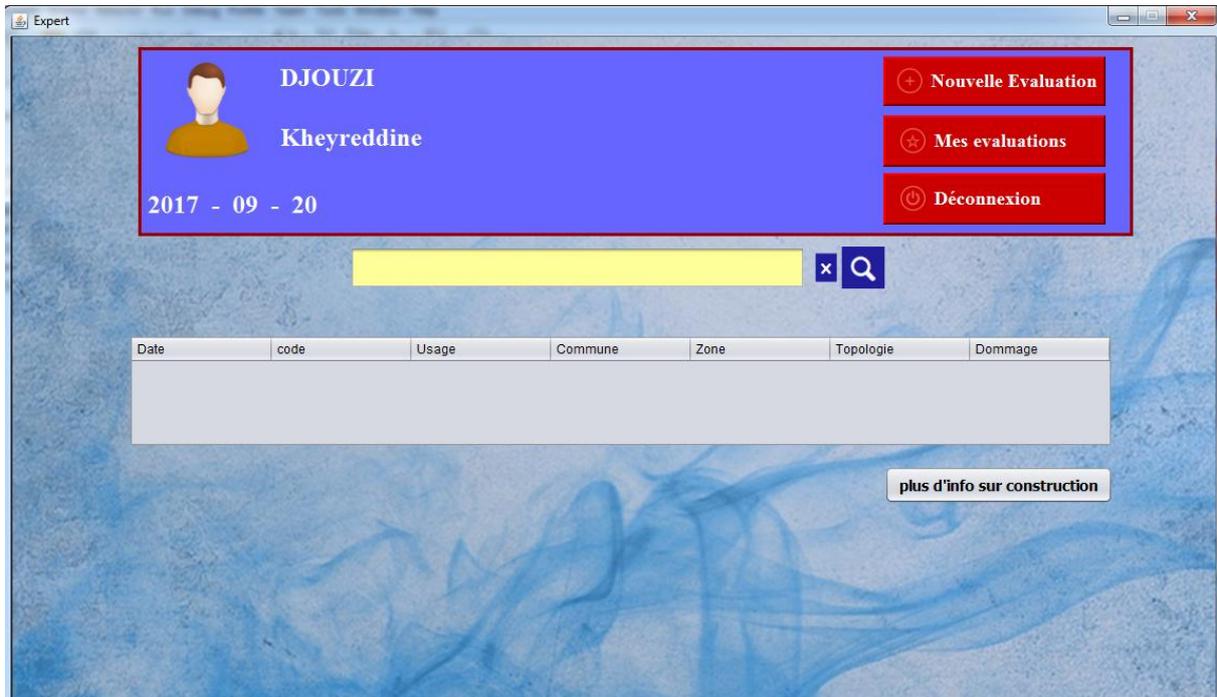


Figure 34: L'interface Expert.

### II.4. L'interface Recherche :

L'utilisateur du logiciel (administrateur, expert ou propriétaire) peut faire une recherche sur tous les constructions déjà évaluées selon plusieurs critères, à travers cette interface.

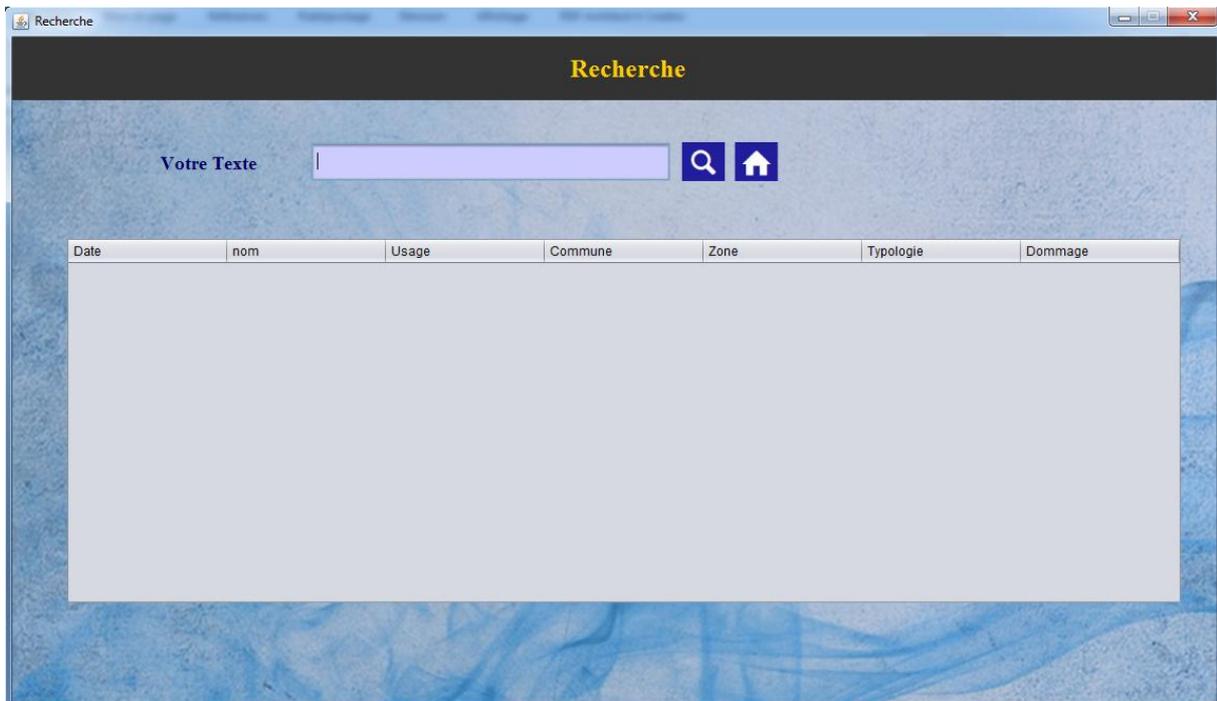


Figure 35: Interface de recherche.

### III. Les tests :

Après avoir réalisé notre logiciel, nous allons maintenant le tester avec un cas réel. Pour cela nous avons introduit les informations d'une fiche d'évaluation des dommages du séisme de Boumerdès 2003 (i.e. une construction déjà évaluée avec la fiche). Cette fiche sera introduite dans l'annexe B.

Commençons tout d'abord par les renseignements généraux sur la construction.

2017-09-20

## Renseignements généraux sur la construction

Wilaya: Boumerdès

Commune: Naciria **2B**

Usage: Logement

Nombre de niveau: 3

Type de typologie: Poteaux poutres

Type de toiture: Terrasse

Type de fondation: Semelle isolé

Construction calculée au séisme: oui

Construction contrôlée par le CTC: oui

Vide sanitaire: non

Sous Sol: non

Accueil Valider

Figure 36: Formulaire de renseignements généraux sur la construction.

Ensuite nous allons remplir le formulaire «Problèmes du sol et constructions adjacentes».

## Probleme autour de la construction

### Problème de sol

Faille: non

Glissement: non

Affaissement: non

Soulevemnt: non

Liquefaction: non

### Influence des constructions adjacentes

La construction menace une autre construction: oui

La construction est menacée par une autre construction: oui

La construction peut etre soutien par une autre construction: oui

La construction peut etre soutenue par une autre construction: oui

Accueil Valider

Figure 37: Formulaire problèmes du sol et constructions adjacentes.

Une fois cela est fait, l'interface « Fondations et infrastructure » s'affiche.

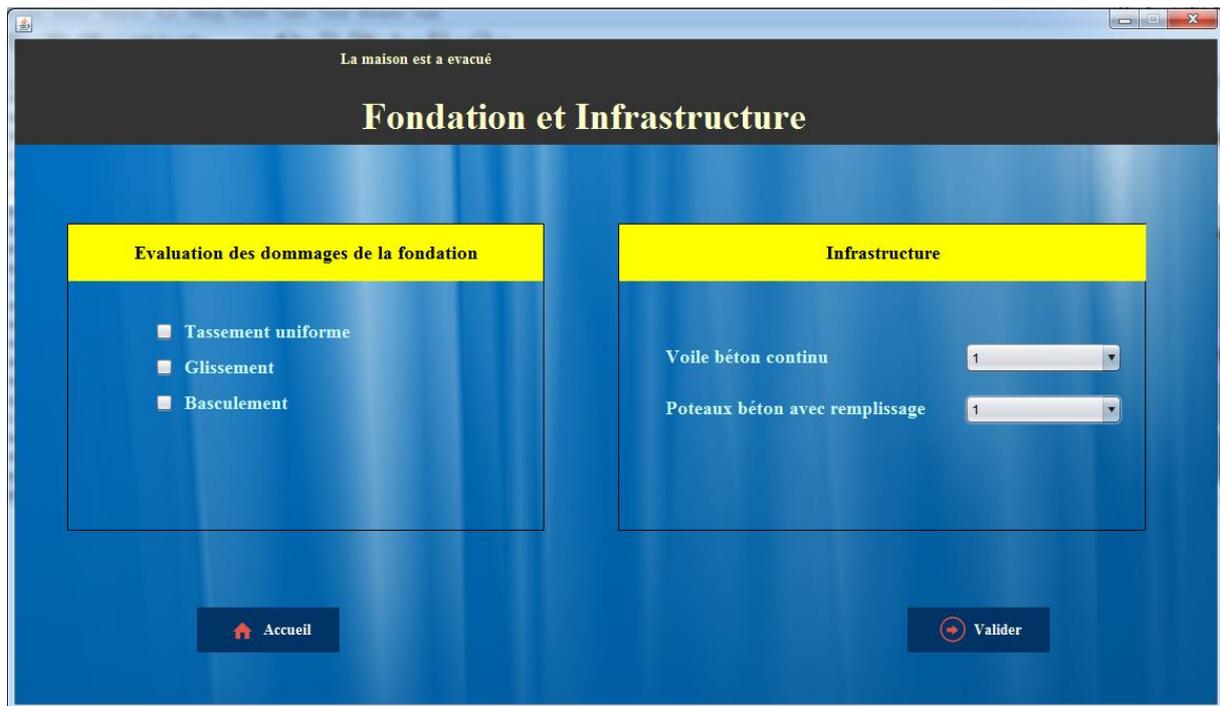


Figure 38: Interface "Fondations et infrastructure".

Et on termine notre test, avec l'évaluation des éléments de la structure résistante puis secondaire.



Figure 39: Interface "Structure résistante".

La maison a évacué ( probleme autour de la construction )

## Elements Secondaire

Escaliers	Autres elements intérieurs	Remplissages extérieurs	Elements extérieurs
<input checked="" type="checkbox"/> Béton <span style="float: right;">4 ▾</span>	<input checked="" type="checkbox"/> Plafonds <span style="float: right;">1 ▾</span>	<input checked="" type="checkbox"/> Maçonnerie <span style="float: right;">2 ▾</span>	<input checked="" type="checkbox"/> Balcons <span style="float: right;">1 ▾</span>
<input type="checkbox"/> Métal <span style="float: right;">▾</span>	<input checked="" type="checkbox"/> Cloisons <span style="float: right;">1 ▾</span>	<input type="checkbox"/> Béton préfabriqué <span style="float: right;">▾</span>	<input checked="" type="checkbox"/> Garde corps <span style="float: right;">1 ▾</span>
<input type="checkbox"/> Bois <span style="float: right;">▾</span>	<input checked="" type="checkbox"/> Element vitrés <span style="float: right;">1 ▾</span>	<input type="checkbox"/> Bardages <span style="float: right;">▾</span>	<input type="checkbox"/> Au vent <span style="float: right;">▾</span>
	<input type="checkbox"/> Autres <span style="float: right;">▾</span>	<input checked="" type="checkbox"/> Acrotère - Corniche <span style="float: right;">1 ▾</span>	
	<input style="width: 100px;" type="text" value="1"/>		<input style="width: 100px;" type="text" value="1"/>

🏠 Accueil
➔ Valider

**Figure 40: Interface "Structure secondaire".**

Une fois l'évaluation est terminée, le système nous retourne le résultat global de cette construction (i.e. le niveau de dommage global de la construction ainsi que la couleur à utiliser).

## Evaluation finale de la construction

[🏠](#)

*le niveau de dommage de cette construction est : 4*

***Dommages importants :***

- \* Dommages non structuraux très importants.
- \* Dommages structuraux considérables.
- \* Fissures en X dans les voiles de contreventement.
- \* Eclatement des noeuds

**Figure 41:Interface "Résultat".**

**IV. Conclusion :**

Dans ce chapitre, nous avons présenté la mise en œuvre de notre logiciel. En premier lieu, nous avons cité L'environnement de développement et les différents outils et logiciels qui nous ont permis de réaliser notre système, puis en second lieu nous avons présenté quelques interfaces représentant les différentes fonctionnalités de ce dernier, et enfin nous avons testé notre logiciel avec des jeux d'essai sur lesquels nous avons abouti à des résultats satisfaisants.

---

# *Conclusion Générale*

---

### *Conclusion générale*

Le travail réalisé dans le cadre de ce mémoire consiste en la conception et la réalisation d'un système expert pour l'évaluation des dommages post-sismiques. L'objectif recherché est la mise au point d'un outil logiciel qui, à partir d'un ensemble de faits (informations sur la construction diagnostiquée) décrivant un cas réel déduit de nouveaux faits (évaluation des différents composants de la construction puis son évaluation finale) à la manière d'un expert humain et ce, en exploitant une base contenant les connaissances nécessaires.

Après avoir entamé l'analyse et la conception de notre système en utilisant « UML », nous avons terminé par la présentation des différents outils de développement utilisés lors de la mise en œuvre de notre application ainsi que la description de son fonctionnement illustré par quelques interfaces.

Tout au long de la réalisation de notre travail, nous avons grandement apprécié l'intérêt de réaliser un système expert. Ce travail nous a en outre permis de :

- Nous introduire un peu dans le domaine d'évaluation des dommages post-sismiques et du génie parasismique ;
- Nous initier au domaine de l'IA. Nous avons ainsi acquis de nouvelles connaissances en la matière, et notamment en « Systèmes Experts » ;
- Apprendre les techniques et les méthodes de conception en particulier la modélisation orientée-objets par « UML » ;
- Étendre nos connaissances et pratiquer la programmation Orientée Objets, en particulier la programmation en Java sous NetBeans;
- Consolider nos connaissances acquises en programmation logique avec SWI-Prolog ;
- Découvrir la JPL et la création d'applications hybrides.
- Nous nous sommes en particulier familiarisés avec les pilotes JDBC pour la connexion entre Java et les BDD, et avons conforté nos connaissances sur l'utilisation des SGBDR (en particulier MySQL).

Bien que tous nos objectifs initiaux soient atteints, notre système peut cependant être amélioré en y apportant les caractéristiques suivantes :

- L'extension du système à d'autre structure autre que celle en maçonnerie pour le cas d'un utilisateur simple (Propriétaire) ;
- La centralisation de la base de données ;
- L'ajout d'un module à la décision pour le confortement des constructions endommagées par le séisme ;
- L'implémentation d'un analyseur sémantique, pour une vérification sémantique des données introduites par l'administrateur, dans le module d'acquisition de connaissances ;

- L'ajout d'un module économique pour le calcul des couts de la réhabilitation (l'indemnisation) ;
- L'ajout d'un module pour le recensement des constructions endommagées par le sinistre et ses statistiques.

Enfin, nous souhaitons que ce modeste travail servira d'outil logiciel d'assistance pour les ingénieurs du génie civil dans le cadre de l'expertise des constructions et de support de documentation pour tous ceux qui s'intéresseront aux systèmes experts et leurs applications.

*Bibliographie*  
*Et*  
*Webographie*

---

## Bibliographie

- [1] : K.AOUALI, et al, « conception et réalisation d'un système expert d'aide au diagnostic de l'insuffisance rénale », mémoire de master en informatique, dirigé par M<sup>me</sup> F.AOUGHLIS, faculté de génie électrique et informatique UMMTO, promotion 2007-2008.
- [2] : C.ARIB, et al. «Conception et réalisation d'un système expert d'aide à la production en chimie industrielle et domestique », mémoire de master en informatique, dirigé par M<sup>f</sup> S.KHEMLICHE, faculté de génie électrique et informatique UMMTO, promotion 2009-2010.
- [3] : R.CAFERRA , « Logique pour l'informatique et pour l'intelligence artificielle », edition hermes-lavoisier 2011
- [4] : J.CONALLEN « Modélisation des applications web avec UML» Edition Eyrolles, 2000.
- [5] : N.DAMMOUCHE, et al, « Conception et réalisation d'un système expert d'aide au diagnostic des infections dentaires. », mémoire de master en informatique, dirigé par M<sup>f</sup> S.KHEMLICHE, faculté de génie électrique et informatique UMMTO, promotion 2008-2009.
- [6] : C.DELANNOY, »Programmer en Java », edition eyrolles 2008.
- [7] : C-H.DOMINÉ, »Techniques de l'intelligence artificielle », édition DUNOD, janvier 1988.
- [8] : J-M.DOUDOUX, « développons en java », mars 2016, En ligne : « [www.jmdoudoux.fr/java/dej/dej\\_2\\_10.pdf](http://www.jmdoudoux.fr/java/dej/dej_2_10.pdf) », lien valide le 21-09-2017
- [9] : H.FARRENY, « Les systèmes experts : principes et outils », BERTI ÉDITION 1992.
- [10] : J.GABAY, et al, « UML 2 analyse et conception : Mise en œuvre guidée avec études de cas Tome 1 », Edition DUNOD, avril 2008.
- [11] : S.GRAINE, « Les systèmes experts », édition l'abeille, 2002.
- [12] : S.GRAINE «UML 2 pour une modélisation orientée objet» Edition l'abeille, 2009.
- [13] : B.HADID, « conception et réalisation d'un système expert d'aide au diagnostic des pannes automobiles »mémoire de licence en informatique, dirigé par M<sup>me</sup> F.AMIROUCHE, faculté de génie électrique et informatique UMMTO, promotion 2014-2015.
- [14] : E.KIANGALA, « Réalisation d'un système expert pour le diagnostic et la thérapeutique de la maladie de la lèpre », Insitut Superieur de Techniques Appliquées (ISTA) Kinshasa - Ingénieur en Genie Electrique orienté Informatique Appliquée 2011. En ligne : « [http://www.memoireonline.com/08/13/7290/m\\_Realisation-dun-systeme-expert-pour-le-diagnostic-et-la-therapeutique-de-la-maladie-de-la-lepre.html](http://www.memoireonline.com/08/13/7290/m_Realisation-dun-systeme-expert-pour-le-diagnostic-et-la-therapeutique-de-la-maladie-de-la-lepre.html) », lien valide le 21-09-2017

[15] : J.RUNBAUGH, et al, « The Unified Modeling Language User Guide », Addison-wesley, 2005 .

[16] : A.THAYSE, et al, « Approche Logique de l'intelligence artificielle : 1. De la logique classique à la programmation logique », edition DUNOD 1990.

[17] : J.WIELEMAKER, “swi prolog 7.4 Reference Manual”, avril 2017, En ligne : « [www.swi-prolog.org/download/stable/doc/SWI-Prolog-7.4.1.pdf](http://www.swi-prolog.org/download/stable/doc/SWI-Prolog-7.4.1.pdf) », lien valide le 21-09-2017

[18] : Règlement parasismique algérien, 2003

## *Webographie*

[19] : <http://www.projet-plume.org/fr/fiche/netbeans>, lien valide le 21-09-2017.

[20] : Le site officiel de l'UML « [www.uml-diagrams.org](http://www.uml-diagrams.org) », lien valide le 21-09-2017.

[21] : <http://img.lelivrescolaire.fr/svt-cycle-4-2017/les-risques-sismiques-et-volcaniques/qu-est-ce-qu-un-seisme-et-comment-s-en-premunir-/3000.svt.2.inf1.png>, lien valide le 21-09-2017.

[22] : <http://jpl7.org/740/doc/index.html>, lien valide le 21-09-2017.

[23] : <https://fr.wikipedia.org/wiki/MySQL>, lien valide le 21-09-2017.

[24] : <http://www.wampserver.com/>, lien valide le 21-09-2017.

[25] : <https://fr.wikipedia.org/wiki/PhpMyAdmin>, lien valide le 21-09-2017.

*Annexe*

A

---

## **I. Définition de l'UML : [4]**

UML (Unified Modeling Language), que l'on peut traduire par « langage de modélisation unifié », est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, il est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

UML n'est pas une méthode, dans la mesure où elle ne représente aucune démarche. A ce titre, UML est un formalisme de modélisation objet. Le mot méthode parfois utilisé par abus de langage ne doit pas donc être entendu comme une démarche.

## **II. Un peu d'histoire : [12]**

Au début des années 90, une cinquantaine de méthodes objets ont vu le jour. Ce qui est un signe de l'intérêt du sujet, et également de confusion. Toutes ces méthodes utilisaient à peu près les mêmes concepts de classes, d'associations, de partition en sous-systèmes...

En octobre 1994, Grandy Booch et Jim Rumbaugh fondèrent le Rational Software Corporation pour unir leurs efforts en vue de créer une norme industrielle unique à partir de leurs méthodes. C'est ainsi que née la méthode unifiée (Unified Method) en octobre 1995. Ils sont ensuite rejoints par Ivar Jacobson, l'inventeur des cas d'utilisation (Uses Cases), et ont publiés UML 0.9 en juin 1996, puis UML 1.0 en janvier 1997.

La version 1.1, mise au point avec d'autres partenaires, devient un standard en novembre 1997 lorsqu'elle est acceptée par l'OMG (Object Management Group). La poursuite du développement d'UML fut ensuite intégralement remise aux mains de l'OMG. En juillet 1998, l'OMG publiait l'UML 1.2, puis UML 1.3 en juin 1999. En mai 2002, UML 1.4, qui présentait de petites améliorations et quelques extensions, voyait le jour. UML 1.5 a été publié en mars 2003 avec également quelques correctifs.

UML 2.0, publié en 2005, présente quant à lui de profondes modifications et de véritables extensions par rapport aux versions 1.x, comme par exemple le diagramme d'activité et le diagramme de séquence.

La version courante de l'UML est 2.5, cette dernière à été publiée en juin 2015.

---

### III. Présentation générale des diagrammes : [20]

UML dans sa version 2.5 propose vingt-trois diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles :

#### III.1. Les diagrammes structurels ou statiques :

Ces diagrammes, au nombre de douze, ont vocation à représenter l'aspect statique d'un système (classes, objets, composants...) :

- « Class diagram », Diagramme de classe ;
- « Object diagram », Diagramme d'objet ;
- « Package diagram », Diagramme de paquetage ;
- « Model diagram » ;
- « Composite structure diagram », Diagramme de structure composite ;
- « Internal structure diagram » ;
- « Collaboration use diagram » ;
- « Component diagram », Diagramme de composant ;
- « Manifestation diagram » ;
- « Deployment diagram », Diagramme de déploiement ;
- « Network architecture diagram » ;
- « Profile diagram ».

#### III.2. Les diagrammes de comportement :

Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Onze diagrammes sont proposés par UML :

- « Use case diagram », Diagramme des cas d'utilisation ;
- « Information flow diagram » ;
- « Activity diagram », Diagramme d'activités ;
- « State machine diagram » ;
- « Behavioral state machine diagram » ;
- « Protocol state machine diagram » ;
- « Interaction diagram », Diagramme d'interaction ;
- « Sequence diagram », Diagramme de séquence ;
- « Communication diagram », Diagramme de communication (anciennement appelé collaboration).
- « Timing diagram », Diagramme de temps.
- « Interaction overview diagram » ;

## IV. Les diagrammes utilisés pour la conception de notre système :

### IV.1. Diagramme de cas d'utilisation : [20]

#### IV.1.A. Définition :

Formalisés par Ivar Jacobson, un cas d'utilisation (en anglais use case) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le format de représentation d'un cas d'utilisation est complètement libre mais UML propose un formalisme et des concepts issus de bonnes pratiques.

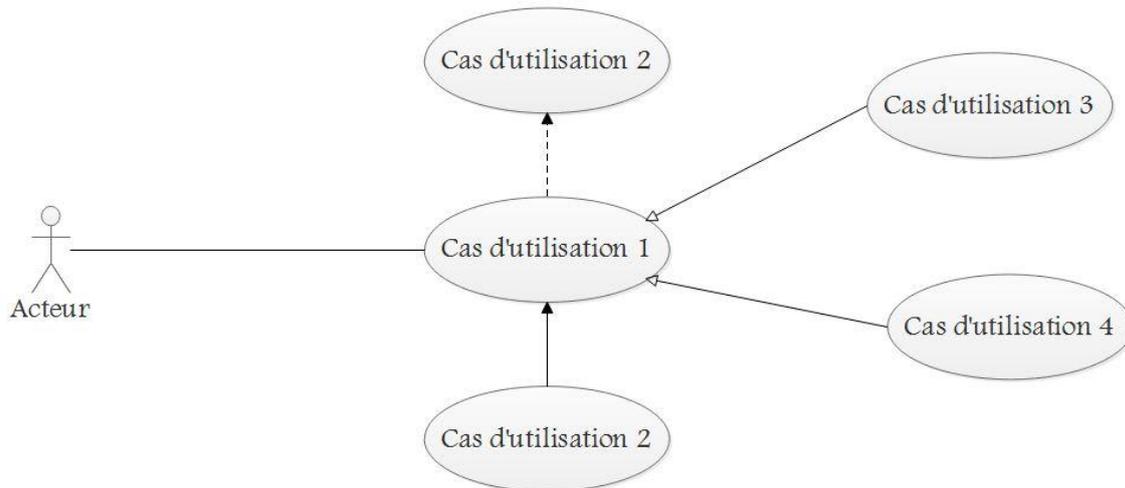
Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, représenté par une boîte rectangulaire, produisant sur un acteur, appelé acteur principal, et ceci indépendamment de son fonctionnement interne.

#### IV.1.B. Éléments de modélisation des diagrammes de cas d'utilisation :

- **L'acteur** : est un ensemble cohérent de rôles, joués par les utilisateurs en interaction avec le cas d'utilisation. Il peut être un humain, une machine ou tout autre système. Il est représenté par un bonhomme en fil de fer (en anglais stick man).
  - **Acteur principal**: regroupe les acteurs qui utilisent les fonctions principales du système. On le représente généralement à gauche (idéalement il y en a un seul).
  - **Acteur secondaire**: regroupe les acteurs qui effectuent les tâches administratives et de maintenances. On le représente généralement à droite.
- **Le cas d'utilisation** : ou un use case est un ensemble de séquences d'actions qu'un système exécute pour produire un résultat tangible par un acteur. Chaque cas d'utilisation doit comporter un nom qui le distingue des autres cas d'utilisation.

Les relations entre les cas d'utilisation :

- « **includ** » : (**Inclusion**) indique que le cas d'utilisation source incorpore explicitement le comportement d'un autre cas d'utilisation à un emplacement précisé par la source.
- « **extend** » : (**Extension**) indique que le cas d'utilisation cible étend le comportement du cas d'utilisation source.
- **Généralisation**: les cas d'utilisation descendants héritent des propriétés de leur parent/ généralisation dans laquelle les objets de l'élément spécialisé (l'enfant) peuvent remplacer les objets de l'élément général (le parent).



**Figure 42: Exemple de diagramme de cas d'utilisation.**

## IV.2. Diagramme de séquence : [15]

### IV.2.A. Définition :

Le diagramme de séquence représente des échanges de messages entre objets. Il permet de représenter un processus de façon simplifiée, en se concentrant sur les échanges entre acteurs ou avec le système d'information.

Le diagramme de séquence permet de visualiser les messages par une lecture de haut en bas. L'axe vertical représente le temps, l'axe horizontal représente les objets qui collaborent. Une ligne verticale en pointillé est attachée à chaque objet et représente sa durée de vie.

### IV.2.B. Éléments de modélisation des diagrammes de séquences :

- **Ligne de vie** : qui représente l'ensemble des opérations exécutées par un objet, il existe 3 objets :
  - **Les objets d'interfaces** :  
Un objet d'interface représente l'interface entre l'acteur et le système tels que les pages web ou les écrans de saisie.

L'icône : 

- **Les objets entité** :  
Ce sont des objets décrits dans un cas d'utilisation et qui se trouvent dans d'autres cas d'utilisation tels que l'utilisateur.

L'icône : 

- **Les objets de contrôle** :  
Ils représentent les activités des processus du système, ils dirigent les activités des entités et interfaces.

L'icône : 

- **Périodes d'activation** : qui correspond au temps pendant lequel un objet effectue une action. Les périodes d'activations, sont modélisées par des boîtes rectangulaires sur la ligne de vie ;
- **Messages** : les envois sont matérialisés par des flèches étiquetés du nom du message envoyé.

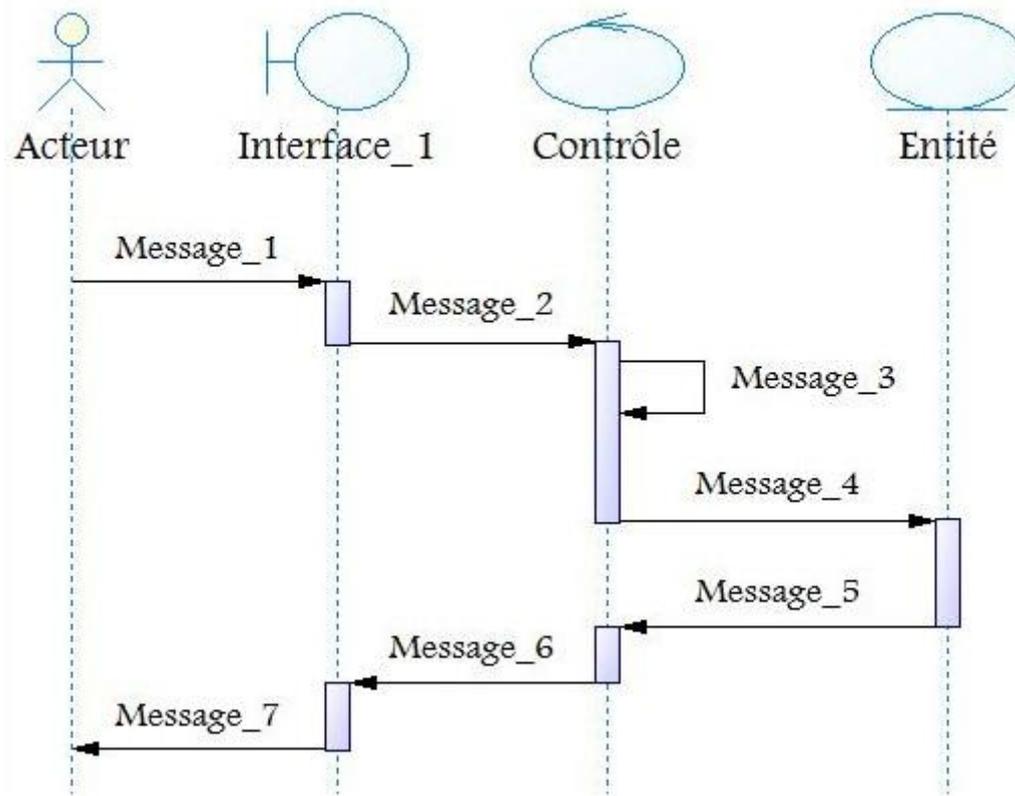


Figure 43: Exemple de diagramme de séquence.

### IV.3. Diagramme de classe : [15]

#### IV.3.A. Définition :

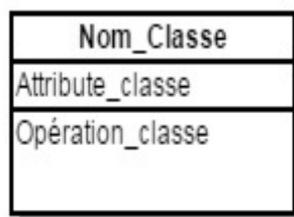
Les diagrammes des classes représentent un ensemble de classes, d'interfaces, de collaboration, et des associations qui relient ses différentes classes. Ce sont les diagrammes les plus utilisés dans la modélisation des systèmes orientés objet. Ils présentent la vue de conception statique des systèmes. Les diagrammes de classes permettent de représenter l'ensemble des informations formalisées, qui sont regroupées dans des classes.

#### IV.3.B. Les notions utilisées par le diagramme de classes :

- **la notation de classe** : une classe est une description abstraite (condensée) d'un ensemble d'objets du domaine de l'application : elle définit leur structure, leur comportement et leurs relations.

Les classes sont représentées par des rectangles compartimentés :

- le 1er compartiment représente le nom de la classe ;
- le 2ème compartiment représente les attributs de la classe ;
- le 3ème compartiment représente les opérations de la classe.

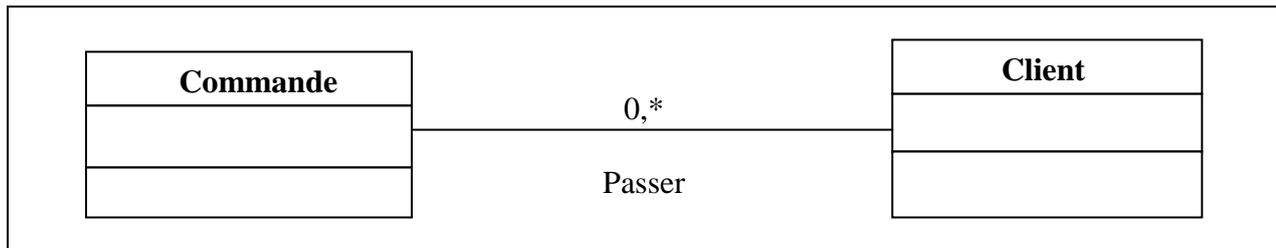


**Figure 44: Représentation graphique d'une classe.**

- **la notation d'attribut :** Une classe correspond à un concept global d'information et se compose d'un ensemble d'informations élémentaires, appelées attributs de classe. Un attribut représente la modélisation d'une information élémentaire représentée par son nom et son format.
  - **la notation d'opération :** l'opération représente un élément de comportement des objets, défini de manière globale dans la classe. Une opération est une fonctionnalité assurée par une classe. La description des opérations peut préciser les paramètres d'entrée et de sortie ainsi que les actions élémentaires à exécuter.
  - **la notation de relation :** s'il existe des liens entre objets, cela se traduit nécessairement par des relations qui existent entre leurs classes respectives. Les liens entre les objets doivent être considérés comme des instances de relations entre classes.
- **Les relations entre les classes :**
    - **L'association :** L'association est la relation la plus courante et la plus riche, elle relie plusieurs classes entre elles. L'association existe entre les classes et non entre les instances elle est introduite pour montrer une structure et non pour montrer des échanges de données.
      - **La multiplicité :** elle définit le nombre d'instances de l'association pour une instance de la classe. La multiplicité est définie par un nombre entier ou un intervalle de valeurs. La multiplicité est notée sur le rôle (elle est notée à l'envers de la notation MERISE).

Notation	Désignation
1	Un et un seul
0,1	Zéro ou un
N,*	De N (entier naturel) à plusieurs
M, N	De M à N (entiers naturels)
0,*	De zéros à plusieurs
1,*	De 1 à plusieurs

**Tableau A.1: Les différentes multiplicités d'une association.**

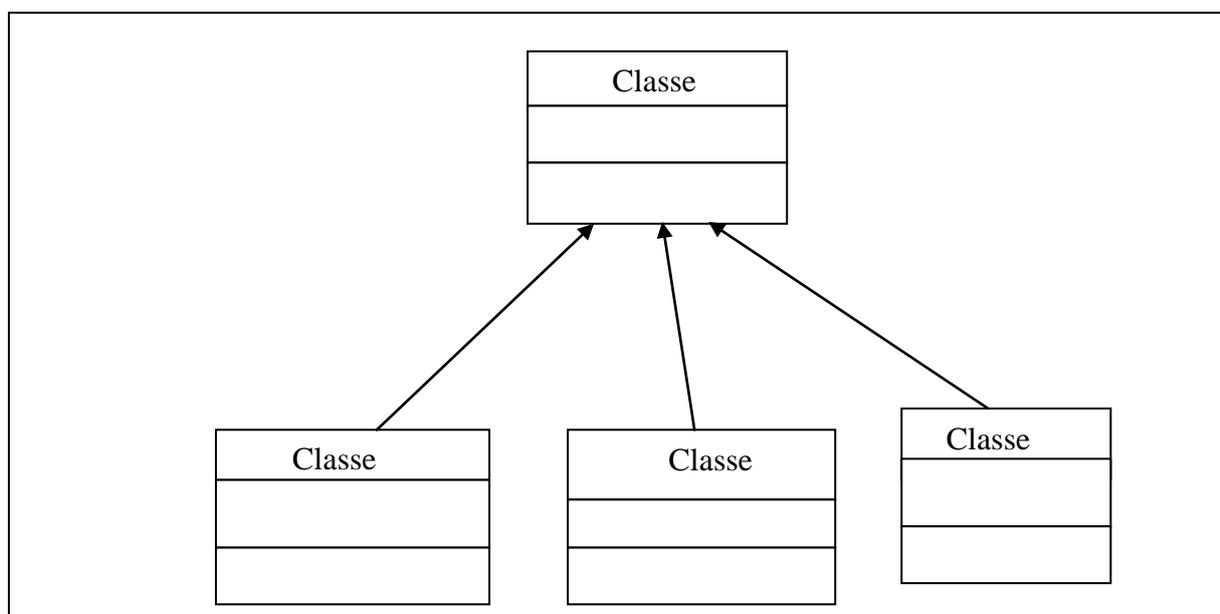


**Figure 45: Relation d'association.**

- **la généralisation/spécialisation** : Le principe de généralisation/spécialisation permet d'identifier parmi les objets d'une classe (générique) des sous-ensembles d'objets (des classes spécialisées) ayant des définitions spécifiques. La classe la plus spécifique (appelée aussi classe fille, classe dérivée, classe spécialisée, classe descendante ...) est cohérente avec la classe la plus générale (appelée aussi classe mère, classe générale ...), c'est-à-dire qu'elle contient par **héritage** tous les attributs, les membres, les relations de la classe générale, et peut contenir d'autres.

Les relations de généralisation peuvent être découvertes de 2 manières :

- la généralisation : il s'agit de prendre des classes existantes déjà mises en évidence et de créer de nouvelles classes qui regroupent leurs parties communes ; il faut aller du plus spécifique au plus général.
- La spécialisation : il s'agit de sélectionner des classes existantes (déjà identifiées) et d'en dériver des nouvelles classes plus spécialisées, en spécifiant simplement les différences.



**Figure 46: Relation de généralisation.**

- **La composition** : La composition est un cas particulier de l'agrégation dans laquelle la vie des composants est liée à celle des agrégats. Elle fait souvent référence à une contenance physique. Dans la composition l'agrégat ne peut être multiple. La composition implique, en plus de l'agrégation, une coïncidence des durées de vie des composants : la destruction de l'agrégat (ou conteneur) implique automatiquement la destruction de tous les composants liés.



**Figure 47: Relation de composition.**

*Annexe* **B**

---

Wilaya de Boumerdès						
<u>Fiches d'évaluation des dommages</u>						
<u>séisme du 21 mai 2001</u>						
code inspecteur :						
date :						
identification de la construction :						
Secteur	zone	Construction calculée au séisme : oui			non	
Adresse ou éléments d'identification			Construction contrôlée : oui			non
Usage de la construction(*)						
Logement	scolaire	commerciale				
Administratif	hospitalier	industriel				
Socioculturel	sportif	réservoir d'eau				
Autre (à préciser)						
Description sommaire						
Age approximatif			vide sanitaire :		oui	non
Nombre de niveaux			Sous-sol :		oui	non
Nombre de joints de dilatation			élément extérieur indépendant			
en elevation			<u>(escalier, auvent, passage, couvert)</u>			
infrastructure						
Problème de sol autour de la construction						
Faïlle :	oui	non	affaissement	soulèvement	oui	non
Liquéfaction :	oui	non	glissement		oui	non
Fondation-infrastructure						
<u>Fondation</u>						
Type de fondation :			infrastructure (dans le cas VS ou S/SOL)			
Type de dommage :			voile béton continu		1 2 3 4 5	
Tassement uniforme :	oui	non	poteaux béton avec remplissage		1 2 3 4 5	
Glissement :			oui	non		
Basculement :			oui	non		
Structure résistante						
Elément porteurs (charges verticales)			<u>Elément de contreventement</u>			
Murs en maçonnerie	1 2 3 4 5		Murs en maçonnerie		1 2 3 4 5	
Voile béton	1 2 3 4 5		Voile béton		1 2 3 4 5	
Poteau métallique	1 2 3 4 5		Portiques béton armé		1 2 3 4 5	
Poteau bois	1 2 3 4 5		Portique métalliques		1 2 3 4 5	
Autre	1 2 3 4 5		Palées triangulées		1 2 3 4 5	
			Autre		1 2 3 4 5	
<u>Plancher-toiture terrasse</u>			<u>Toiture inclinée</u>			
Béton armé	1 2 3 4 5		Charpente métallique		1 2 3 4 5	
Solives métallique	1 2 3 4 5		Charpente bois		1 2 3 4 5	
Solive bois	1 2 3 4 5		Couverture tuile		1 2 3 4 5	
			Couverture amiante ciment		1 2 3 4 5	
			Couverture métallique		1 2 3 4 5	

<u>Elément secondaires</u>						
<u>Escaliers</u>		<u>Remplissages extérieurs</u>				
Béton	1 2 3 4 5	Maçonnerie	1 2 3 4 5			
Métal	1 2 3 4 5	Béton préfabriqué	1 2 3 4 5			
Bois	1 2 3 4 5	Bardages	1 2 3 4 5			
		Autres	1 2 3 4 5			
<u>Autres éléments intérieurs</u>		<u>Éléments extérieurs</u>				
Plafonds	1 2 3 4 5	Balcons	1 2 3 4 5			
Cloisons	1 2 3 4 5	Garde cops	1 2 3 4 5			
Élément vitrés	1 2 3 4 5	Au vent	1 2 3 4 5			
		Acrotère-corniche	1 2 3 4 5			
		Autres	1 2 3 4 5			
<u>Influence des constructions adjacentes</u>						
La construction menace une autre construction		oui	non			
La construction est menacée par une autre construction		oui	non			
La construction peut être soutenue par une autre construction		oui	non			
La construction peut être soutenue par une autre construction		oui	non			
<u>Victimes</u>						
Oui – non – peut être		si oui combien ??				
<u>Commentaire sur la nature et la cause probable des dommages</u>						
	<u>Sens transversal</u>			<u>Sens longitudinal</u>		
Symétrie en plan	bon	moyen	mauvais	bon	moyen	mauvais
Régularité en élévation	bon	moyen	mauvais	bon	moyen	mauvais
Redondances des files	bon	moyen	mauvais	bon	moyen	mauvais
<u>Autres commentaires</u>						
<u>Evaluation finale</u>						
<u>Niveau général des dommages</u>				<u>Couleur à utiliser</u>		
1 - 2 - 3 - 4 - 5				vert – orange - rouge		
<u>Mesures immédiates à prendre :</u>						