

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جامعة عمّار ثليجي بالأغواط
UNIVERSITE AMAR TELIDJI LAGHOUAT
كلية العلوم
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

Mémoire de MASTER

Domain : Mathématiques et Informatique
Filière : Informatiques
Option : Systèmes d'Information et de Décision

Par : BEKADDOUR Mohammed Achraf

THEME

LA CONCEPTION D'UN OUTIL DE DIAGNOSTIQUE DES PANNES INFORMATIQUES

Soutenu publiquement devant le jury composé de:

Mr	L. Chellama	M.A.(A)	Président
Mr	Y. Guellouma	M.A.(A)	Examineur
Mr	B. Ziani	M.A.(A)	Examineur
Mme	B. Kerrouche	M.A.(A)	Encadreur

Année Universitaire 2015/2016

Résumé

Tout ce qu'un utilisateur d'ordinateur non expert en informatique peut rencontrer comme problèmes (virus, perte de fichiers, écran noir...) qui constitue le domaine de travail d'un technicien de maintenance en informatique.

L'objectif de la maintenance informatique est de résoudre rapidement et de manière efficace les problèmes. L'utilisateur doit être amené à réparer ou changer des pièces défectueuses, mais il doit avant tout réaliser un diagnostic pour trouver la solution la mieux adaptée. Pour ce faire il peut l'effectuer par un système installé sur l'ordinateur.

Généralement ce type de systèmes est basé sur les notions d'intelligence artificielle et les notions d'aide à la décision. Notre travail présente la modélisation et le développement de système d'évaluation des pannes informatiques. Nous présentons l'approche utilisée et la modélisation réalisée (graphique et textuelle) par des arbres de décision et en intégrant une combinaison de deux notions, les systèmes experts et les agents conversationnels.

Notre système « ExpertBot Repair » offre une communication (utilisateur-système) efficace. Ce système peut faire le diagnostic d'une manière rapide et sans le recours à un expert en informatique.

Les mots clés: L'évaluation des pannes informatiques, Les systèmes experts, Les agents conversationnels, Les arbres de décisions, L'intelligence artificielle, L'aide à la décision, ExpertBot Repair.

المخلص

يمكن لأي مستخدم كمبيوتر غير مختص في مجال الإعلام الآلي أن يواجه مشاكل (الفيروسات، فقدان ملف، الشاشة السوداء ...) وهي مجال العمل لتقني صيانة الكمبيوتر. صيانة أجهزة الكمبيوتر تهدف إلى حل المشاكل بسرعة وبفعالية ما يتطلب من المستخدم إصلاح أو استبدال الأجزاء التالفة. ولكن يجب عليه أن يقوم أولاً بعملية التشخيص للعثور على الحل المناسب. يمكنه القيام بهاته العملية بواسطة نظام مثبت على جهاز الكمبيوتر. عموماً يستند هذا النوع من النظم على مفاهيم الذكاء الاصطناعي ومفاهيم دعم اتخاذ القرار.

يعرض عملنا نمذجة وتطوير نظام تقييم أعطال جهاز الكمبيوتر. نقدم في هذا المنهج النمذجة (رسمية ونصية) و التنفيذ و ذلك بالمزج بين اثنين من الأنظمة الأنظمة الخبيرة ، وروبوت الدردشة. يوفر نظام «ExpertBot Repair» التواصل بين النظام والمستخدم بصفة فعالة. هذا النظام يمكن أن يقوم بتشخيص الأعطال بطريقة سريعة ودون اللجوء إلى خبير صيانة أجهزة الكمبيوتر.

الكلمات المفتاحية : تقييم أعطال الكمبيوتر , الأنظمة الخبيرة , روبوت الدردشة , شجرة القرار , الذكاء الاصطناعي

, دعم إتخاذ القرار , ExpertBot Repair .

Abstract

All a non-expert computer users may experience such problems (viruses, loss files, black screen ...) which are the area of work of a computer service technician. The goal of computer maintenance is to solve quickly and efficiently computer problems. The user should be required to repair or replace defective parts and components.

But it must first to do a diagnosis to find the best solution. To do this operation he can do it by a system installed on the computer. Generally this type of systems are based on the concepts of artificial intelligence and the concepts of decision tree.

Our work presents the modeling and development of the evaluation system of computer failures. We present the used approach and the modeling performed (graphical and textual) by a combination of two concepts expert systems, and conversational agents. Our "ExpertBot Repair" system offers an effective communication (user system). This system can make the diagnosis in a rapid manner without the support of a computer expert.

Key words: Evaluation of computer failures, Expert systems, Conversational agents, Decision tree, Artificial intelligence, Decision support, ExpertBot Repair.

Remerciements

Je remercie tout d'abord ALLAH, le tout puissant de m'avoir donné la force et la patience et de m'avoir rapproché des personnes qui m'ont soutenu et aidé pour accomplir ce travail.

Mes remerciements s'adressent également à tous les personnes qui ont contribué de près ou de loin avec leurs conseils ou avec leurs encouragements à l'accomplissement de ce travail.

Je tiens à exprimer ma sincère reconnaissance et remerciements à Mme.

KERROUCHE Badra pour l'encadrement, l'aide, l'encouragement et la sympathie qu'il nous a donnés. Grâce à ses conseils, nous avons pu terminer et compléter nos travaux,

Mes remerciements vont particulièrement aux enseignants et administrateurs du d'département de mathématique et informatique.

*Enfin, j'exprime mes vifs remerciements à toute ma famille et spécialement à **mes parents**, que je leurs souhaite une longue vie pleine de bonheur, de santé et de prospérité, c'est à eux que je dois tout. ”*

*BEKADDOUR Mohammed Achraf
mai 2016*

Dédicace

Ce travail est spécialement dédié à :

Ma mère pour toute l'affection qu'elle me procure.

Mon père pour ses innombrables et précieux conseils.

Mes frères et ma sœur pour leur soutien.

Mes très chers amis : Abderazzak, Bouamama, Abdellmadjid, Ossama, badreeddine, Ayoub, zkaria, Hamza Abdelaaziz et tous les autres.

Tous mes camarades.

Toute ma famille, mes oncles, mes tantes, mes cousins et mes cousines.

Et tous ceux qui m'aiment.

Table des matières

Résumé.....	II
المُلخَص	III
Abstract	IV
Remerciements.....	V
Dédicace.....	VI
Table des matières	VII
Liste des figures	IX
Liste des Tableaux	X
Introduction	1
CHAPITRE I : GENERALITES SUR LA MAINTENANCE INFORMATIQUE	
I.1 Introduction	5
I.2 L'ordinateur.....	5
I.3 Composition matérielle d'un ordinateur	5
I.4 Le logiciel.....	7
I.5 Le Dépannages Informatiques.....	8
I.5.1 Les types de pannes informatiques	9
I.5.2 Exemples d'outils de maintenance.....	10
I.6 Conclusion	11
CHAPITRE II : L'ETAT DE L'ART	
II.1 Introduction	13
II.2 Les techniques de diagnostic des pannes informatiques	13
II.2.1 Techniques basées sur les arbres de décision :	13
II.2.2 Techniques basées sur Les systèmes experts.....	14
II.2.3 Techniques basées sur les Systèmes d'aide à la décision	16
II.2.4 Techniques basées sur Le diagramme d'Ishikawa.....	16
II.2.5 Techniques basées sur les arbres des causes :.....	18
II.2.6 Techniques basées sur des modèles :	18
II.2.7 Techniques basées sur les statistiques	19
II.2.8 Techniques basées sur L'apprentissage automatique :.....	20
II.3 Le ChatBot	21
II.4 Conclusion	23

CHAPITRE III : LA CONCEPTION DE L'OUTIL D'EVALUATION DES PANNES INFORMATIQUES (EXPERTBOT REPAIR)

III.1	Introduction	25
III.2	Aperçu générale du système	25
III.3	La conception de l'outil	27
III.4.1	Le diagramme de cas d'utilisation	27
III.4.2	Les diagrammes de séquence	28
III.4	Le développement du système expert	34
III.4.1	Le système expert	34
III.4.2	La base de faits	34
III.4.3	Description de la base de connaissance	36
III.4.4	Description des règles.....	36
III.4.5	Le moteur d'inférence	37
III.4.6	Exemple	37
III.5	Le développement du chatbot.....	38
III.5.1	Présentation	38
III.5.2	L'Artificial Intelligence Markup Language	40
III.5.3	Exemple	41
III.6	Conclusion	42

CHAPITRE IV : L'IMPLEMENTATION

IV.1	Introduction	44
IV.2	II. Plateformes logicielles	44
IV.3	Les interfaces	46
IV.3.1	Le système Expert Bot Repair :	46
IV.3.2	La phase de choix de mode d'évaluation :	47
IV.3.3	Le mode système expert	48
IV.3.4	Le mode ChatBot.....	49
IV.3.5	L'application d'administrateurs :	50
IV.4	Structure des programmes :	51
IV.4.1	Les codes des fonctions principales de l'application	51
IV.5	Conclusion	55
	Conclusion générale.....	57
	Bibliographie.....	59

Liste des figures

Figure II- 1 Le diagramme d'Ishikawa	17
Figure II- 2 Le diagramme des causes de H.Schyns.....	17
Figure III- 1 L'architecture générale de systèmes ExpertBot Repair	25
Figure III- 2 L'arbre de décision source [34]	35
Figure III- 3 Le schéma illustratif de système ChatBot.....	38
Figure III- 4 L'arbre N-aire	39
Figure III- 5 La structure du langage AIML	40
Figure III- 6 Exemple d'une application de ChatBot	41
Figure III- 7 Le fichier source AIML de l'exemple	42
Figure IV- 1 Le Netbeans IDE	45
Figure IV- 2 Interface graphique de système ExpertBot Repair	46
Figure IV- 3 La fenêtre de choix de système d'évaluation	47
Figure IV- 4 L'interface de système expert	48
Figure IV- 5 L'interface de système ChatBot	49
Figure IV- 6 L'interface de système d'administrateur	50

Liste des Tableaux

Tableau 1 La représentation statique d'un système d'exploitation, en couches.....	8
Tableau 2 la représentation d'un noeud d'arbre de décision	36

INTRODUCTION

Introduction

L'informatique pénètre tous les domaines de notre vie quotidienne et rares sont les activités qui ne soient aujourd'hui marquées de son cachet. Les ordinateurs calculent, écrivent, parlent et chantent. Ce sont des outils de travail et moyens de recherche.

L'ordinateur est sûrement d'une énorme importance dans la vie moderne. Un travail assisté par ordinateur est fort appréciable, car ce dernier offre beaucoup d'avantages : grande capacité de mémorisation, fidélité, rapidité, précision, traitement automatique de l'information...

En qualité de machines, les ordinateurs peuvent être sujets à de nombreux problèmes et pannes qui nécessitent la recherche de solutions.

Divers outils d'évaluation des pannes informatiques existent et sont souvent basés sur la notion de systèmes experts. Ces outils permettent aux utilisateurs de comprendre la cause de la panne et de les guider vers d'éventuelles solutions. Généralement ces outils doivent être dotés d'un outil de communication (utilisateur-système) le plus proche possible de l'évaluation par un technicien de maintenance.

Notre travail consiste en la conception d'un système d'évaluation des pannes informatiques basé sur deux notions de l'intelligence artificielle à savoir les systèmes experts et les agents conversationnels avec une modélisation des bases des connaissances sous la forme d'un arbre de décision.

Motivation

L'intérêt du sujet est qu'il y'a des difficultés d'obtenir des informations utiles sur les pannes touchant l'ordinateur et ses composants. Ces informations sont acquises par l'expérience et avec le nombre limité des techniciens on a besoin de systèmes jouant le rôle d'un expert et que n'importe qu'elle personne peut solliciter ses services. Ce type de système pourra dépanner d'une manière rapide et sans déplacement ni coût certaines pannes informatiques.

Plan de travail

Ce mémoire est organisé en quatre chapitres : Le premier chapitre présente des généralités sur l'informatique et le micro-ordinateur, Dans le deuxième chapitre, nous avons présenté également quelques travaux et projets de recherche en rapport avec notre contribution. Le troisième chapitre est consacré à la modélisation de notre système, il présente l'approche utilisée et la modélisation réalisée (graphique et textuelle). Dans le quatrième chapitre, nous présentons la réalisation et l'implémentation de notre système. Enfin, nous terminons ce mémoire par une conclusion générale dans laquelle nous présentons un bilan du travail effectué ainsi que les perspectives et les travaux futurs méritant d'être effectués pour améliorer et approfondir ce travail.

CHAPITRE I
GENERALITES SUR LA
MAINTENANCE INFORMATIQUE

I.1 Introduction

L'informatique fait aujourd'hui partie intégrante de notre vie. C'est un domaine d'activité scientifique, technique et industriel concernant le traitement automatique de l'information par l'exécution de programmes informatiques par des machines . Le programme est une suite ordonnée d'instructions traduites dans un langage de programmation et capable de résoudre un problème donné.

Depuis ses débuts en 1950, l'informatique a connu des transformations profondes. La transformation des matériels informatiques en vitesse, puissance, fiabilité, miniaturisation est impressionnante. Un ordinateur est un appareil électronique qui traite les informations dans une unité centrale selon un programme enregistré en mémoire. Avec l'utilisation large des ordinateurs, plusieurs pannes peuvent apparaître de type matérielles ou logicielles et alors pour les évaluer il faut connaître c'est quoi un ordinateur et c'est quoi un logiciel.

I.2 L'ordinateur

Un ordinateur est un ensemble d'éléments dont la pièce principale est l'unité centrale. À cette dernière, sont reliés les périphériques : moniteur (écran), clavier, souris, imprimantes, modem, etc. On désigne par le terme de "composant", les éléments qui constituent la partie matérielle de l'unité centrale Le ou les systèmes d'exploitation installés ainsi que les applications forment la partie logicielle. Ce premier chapitre va nous permettre de faire connaissance avec les composants matériels de l'ordinateur ainsi qu'avec leur fonctionnement [1].

I.3 Composition matérielle d'un ordinateur

Un boîtier : il va abriter les composants essentiels de l'ordinateur. Diverses formes et dimensions existent (tour, mini ou moyen tour, à plat, etc...).

L'alimentation : à partir du courant du secteur 220v 50Hz alternatif, elle fournit les tensions continues (5 Volts, 12 Volts, 3,3 Volts, etc.) nécessaires aux divers composants de l'ordinateur

et la puissance en Watt requise, elle est ventilée pour la refroidir. Sur les portables une batterie fait tampon et fournit l'alimentation en absence du courant sur le secteur.

La carte mère : c'est une carte de circuits imprimés avec divers connecteurs, un ou deux "Socket" pour support du ou des processeurs, les 'slots' pour enficher les composants : pavé du bios, cartes, connecteurs ... Elle peut intégrer les fonctions de la carte vidéo, son, réseau, etc.

Le jeu de composants : le 'chipset', gère tous les composants placés sur la carte mère.

Le processeur : c'est la puissance de calcul, un programme lui envoie les données à traiter et il fournit les résultats. Il est de plus en plus puissant et vélocité mais il faut souvent le refroidir par un ventilateur auxiliaire.

La mémoire vive (RAM) : provisoirement elle va stocker les programmes et les données qui vont alimenter le processeur, et engranger les résultats des calculs. Les types de mémoire RAM évoluent (EDO, SDRAM, DDRAM, SDRAM DDR, ...) ainsi que la capacité des barrettes (512 Mo, 1 Go...). Les accès entre la mémoire vive et le processeur sont les plus rapides mais dépendent de la vitesse que peut supporter le bus ou transitent les données. Avec une capacité en mémoire vive importante on évite les accès qui sont plus lents vers le disque dur.

Le Bios : implanté dans une mémoire non volatile le logiciel Bios s'occupe de toute la séquence de démarrage de la machine. Il détecte la mémoire et le processeur, les cartes d'extension, les principaux périphériques, et de plus en plus puissant gère leurs paramètres (économies d'énergie, alarmes de surchauffe, etc.). Il est paramétrable. Le Bios (de marques Award Bios, Amibios, Phoenix Bios...) peut se "flasher" pour le remplacer 1 par une nouvelle version plus performante ou qui reconnaîtra mieux un composant particulier.

Une pile électrique : de type bouton elle permet de sauvegarder l'horloge système pendant l'arrêt de l'ordinateur et la coupure d'alimentation secteur. On peut l'enlever pour remettre le Bios aux valeurs initiales par défaut s'il est vérolé ou la changer si elle est usée (dérive de l'horloge).

Les interfaces (IDE, EIDE, SCSI, Serial ATA...): permettent de connecter des périphériques de stockage sur mémoire de masse via des nappes de fils.

Le disque dur : Il est préalablement formaté par le constructeur (formatage bas niveau). On peut faire un partitionnement et il faut préparer les partitions à recevoir un système de fichiers; ceci est aussi appelé formatage logique (type : Windows FAT, VFAT, NTFS, Linux Ext3 ou Ext4, Mac HFS,...). Il contient les données permanentes : le système d'exploitation, les pilotes, les programmes logiciels et les données semi-permanentes (lecture et écriture).La connectique est de norme IDE/ATA ou Serial ATA ou SCSI ; des boîtiers externes permettent de connecter des disques durs en USB ou FireWire.

Lecteur de CDROM : permet seulement de lire des données (devient à ce titre obsolète) stockées sur des CDROMS ayant une capacité 650 méga-octets. Il permet de lire des CDROMS musicaux via un logiciel de lecture et de restitution spécifique.

Lecteur-Graveur de CDROM : Il permet en plus de lire, de graver par laser des CDROMS et permettre des sauvegardes ou duplications de données. Les CD comme les DVD sont sensibles aux rayures.

Lecteur DVD-ROM : permet de lire des CD de Vidéos (DVD) ou de données. La capacité du DVD est de 4,7 Go Lecteur-Graveur DVD-ROM : permet de lire, de graver par laser des DVD de Vidéos par exemple ou de stockage de données.

Les slots PCI, AGP, PCI Express : sont des connecteurs qui permettent l'en fichage des cartes internes.

Carte vidéo : permet le pilotage et l'envoi des données graphiques vers l'écran. Elles deviennent pour les jeux et la vidéo en 3D hyperréaliste de véritables ordinateurs auxiliaires de calculs avec processeur, mémoire RAM DDR .Elle peut posséder une sortie TV.

Carte son : pilote le son vers des haut-parleurs, permet des entrées microphone ou auxiliaires, peut gérer des effets (sur round...).

Carte modem analogique ou numérique : permet de se relier au réseau téléphonique (pour des fonctions de fax, Minitel, Internet, répondeur...).

Carte réseau : souvent de type Ethernet (10/100Mbps ou Gigabit) sert à relier plusieurs ordinateurs en réseau filaire local (LAN) peut servir de support vers un modem Câble ou ADSL, la connectique classique est de type RJ45, les données transitent en série [2].

I.4 Le logiciel

Les logiciels sont composés de programmes informatiques, qui indiquent à l'ordinateur comment effectuer les tâches. Le logiciel détermine les tâches qu'un appareil informatique peut effectuer. Le Logiciel n'est pas synonyme de programme informatique. Un logiciel est un ensemble typiquement composé de plusieurs programmes, ainsi que tout le nécessaire pour les rendre opérationnels : fichiers de configuration, images bitmaps, procédures automatiques. Les programmes sont sous forme de code binaire ainsi que parfois sous forme de code source.

La plus importante pièce de logiciel est le système d'exploitation. Le système d'exploitation (appelé parfois OS) est le logiciel qui va faire l'interface entre un système physique, comme un ordinateur, une tablette ou un smartphone, et d'autres programmes, qui vont pouvoir interagir avec cette machine. C'est lui qui va gérer les accès aux ressources comme

la mémoire vive, le microprocesseur ou encore l'accès au(x) disque(s). Dans la grande majorité des cas, un programme est conçu pour fonctionner avec un OS en particulier (et même une version précise de cet OS) [3].

Tableau 1 La représentation statique d'un système d'exploitation, en couches.

Programmes d'application
Services communication
Services du système d'exploitation
Pilotes des ressources physiques
Interfaces physiques, noyau exécutif

I.5 Le Dépannages Informatiques

Un dépannage informatique est une opération technique, généralement faite par un professionnel, visant à rétablir le fonctionnement normal de tout équipement informatique, à la suite d'une panne matérielle ou à un usage dommageable d'un des logiciels utilisés.

Le technicien de maintenance informatique (parfois appelé technicien support ou technicien d'exploitation) est chargé de s'assurer du bon fonctionnement des postes de travail, des logiciels et des périphériques (imprimantes, etc.) des utilisateurs et, en cas de panne, être en mesure de les dépanner.

I.5.1 Les types de pannes informatiques

1) Panne matérielle : Même si ce type de panne est de moins en moins fréquent, nul n'est à l'abri d'une surchauffe d'ordinateur, ou d'un défaut du bloc d'alimentation. L'aspect positif de ce type de panne est que les pièces d'un ordinateur sont communes, bon marché, facilement disponibles et interchangeables par un technicien qualifié.

2) Panne logicielle système : La complexité des systèmes d'exploitation actuels est source de nombreuses pannes.

3) Panne logicielle applicative : Les applications spécifiques de l'entreprise peuvent être la cause de pannes logicielles. Pour en limiter l'impact, il est toujours souhaitable que l'entreprise dispose d'un contrat de maintenance avec tous les éditeurs d'applications tierces.

4) Panne de communication : Les Petite et moyenne entreprise ne sont plus rien sans leur système de communication, surtout depuis que la téléphonie s'appuie sur Internet pour assurer le transport de la voix.

5) Panne réseau local : Ce sont plutôt des pannes rares, mais importants car elles sont souvent causées par un câblage rendu défectueux suite à de mauvaises manipulations, comme un arrachement involontaire des raccords. Pour éviter ce type de panne, il est recommandé de faire certifier le câblage informatique par une entreprise qualifiée et de réduire la longueur des câbles de raccord au strict minimum [4].

I.5.2 Exemples d'outils de maintenance

1) Hiren's BootCD

Hiren's BootCD rassemble de nombreux outils spécialement conçus pour régler les problèmes liés au système Microsoft Windows et se présente sous la forme d'un CD bootable. Il est aussi possible de copier le logiciel sur un périphérique de stockage USB. Ce logiciel propose de nombreux antivirus qui permettent d'effectuer des analyses au démarrage. Performants, ces programmes permettent de déceler les menaces potentielles telles que les chevaux de Troie, les vers ou encore les spywares. Hiren's BootCD propose également divers outils permettant de créer des copies de sauvegarde des données présentes sur les disques locaux. Parmi les outils proposés, on peut citer CloneDisk, CopyWipe ou encore DiskImage. Grâce aux outils inclus dans ce programme, l'utilisateur pourra naviguer à travers les fichiers présents sur les disques durs et les éditer même si le système ne démarre pas. L'utilitaire Ext2Explore permet par exemple de naviguer dans les différentes partitions du disque dur [5].

2) UBCD4WIN

UBCD4WIN propose une suite de logiciels gratuits vous permettant de réparer, restaurer, et effectuer un diagnostic complet de votre PC. Vous aurez également à votre disposition divers outils très pratiques. Le but est en effet de démarrer un PC inopérant à partir de ce CD pour tester sa mémoire RAM, lancer la console de récupération Windows, récupérer/restaurer ses données effacées, réparer Windows, effacer un mot de passe oublié, créer un admin, éditer le Registre, récupérer les ruches du Registre, entretenir le disque, restaurer une partition raw, gérer des sauvegardes, partitionner, retrouver des fichiers effacés, chasser les virus, surfer sur internet, servir de bac à sable, défragmenter votre disque dur, modifier les volumes NTFS, etc [6].

UBCD4Win est important au réparateur amateur, un outil à mettre dans toutes les trousse de secours. Ses avantages sont :

- Permet d'intervenir sur une machine dont Windows ne démarre plus.
- Disques SATA reconnus. Drivers spécifiques si nécessaires (plug-ins).
- Accès aux fichiers de sauvegarde du SVI même si formatage NTFS.
- Environnement Windows PE : commandes et programmes Windows utilisables.
- Panneau de langues pour changer de clavier même une fois démarré.
- Fait pour Windows XP, mais reste utile avec Vista et Win7.
- Création possible depuis Vista ou Win7 (il faut disposer d'un CD d'installation XP avec l'i386).
- Création possible depuis un Windows XP non activé (ne pas entrer de serial et poursuivre l'installation).
- Utilitaires portables d'une clé USB.
- Sauvegarde vers l'USB, le réseau.
- Création du CD très automatisée, gravure automatique de l'image ISO débrayable.
- Nombreuses possibilités d'adaptation et d'amélioration.
- Création d'une clé USB au lieu d'une ISO encore plus rapide, facile, et souple (une clé est modifiable après coup).

I.6 Conclusion

Nous avons présenté dans ce chapitre des notions générales sur l'informatique, l'ordinateur et les pannes informatiques. Les pannes matérielles sont pour la plupart difficile à détecter vu le nombre de composants qu'un PC dispose. Il faut donc procéder de manière méthodique, étape par étape pour arriver à cibler l'origine du problème.

Plusieurs techniques et outils ont été développés pour l'évaluation des pannes informatiques c'est ce que nous allons voir dans le deuxième chapitre.

CHAPITRE II :

L'ETAT DE L'ART

II.1 Introduction

Le Diagnostic des pannes informatiques est le processus d'identification des causes de la dégradation du fonctionnement d'un système. Il est basé sur les symptômes observables, à savoir, la détermination de la faute qui a conduit à la défaillance. Un dépannage informatique est une opération technique, visant à rétablir le fonctionnement normal de tout équipement informatique, à la suite d'une panne matérielle ou à un usage dommageable d'un des logiciels utilisés.

Bien que le diagnostic a été historiquement un processus manuel nécessitant une intervention humaine, des techniques automatisées ont fait leur apparition dans de nombreux secteurs, notamment les télécommunications, les services Internet ...etc. Ce chapitre présente une étude sur les techniques de diagnostic des pannes informatiques.

II.2 Les techniques de diagnostic des pannes informatiques

II.2.1 Techniques basées sur les arbres de décision :

Les arbres de décision constituent une méthode récente et efficace d'exploration de données, en vue de la prédiction d'une variable qualitative à l'aide de variables de tout type (qualitatives et/ou quantitatives). Cette flexibilité constitue un avantage par rapport à certains outils de classification, prévus pour des prédicteurs d'un seul et même type. Il s'agit d'une méthode itérative, dite de partitionnement récursif des données. En effet, la méthode construit des classes d'individus, les plus homogènes possible, en posant une succession de questions binaires (de type oui/non) sur les attributs de chaque individu [7].

Contrairement à beaucoup d'outils de classification : régression logistique, SVM (Support Vector Machines), etc., les arbres de décision sont extrêmement intuitifs et fournissent une représentation graphique, parlante et facile à lire, d'un protocole de classification des individus. Cette représentation graphique est sous forme d'un arbre constitué de feuilles terminales (les classes d'individus) obtenues en suivant un chemin le long des nœuds, chaque nœud correspondant à une question binaire utilisant une variable du jeu de données [7].

Mike Chen et al présentent une approche d'apprentissage d'arbre de décision pour le diagnostic des pannes dans les grands sites Internet. Ils enregistrent des propriétés d'exécution de chaque demande et d'appliquer des techniques d'apprentissage automatique et Data Mining automatisé pour identifier les causes et les échecs. Ils évaluent cette approche en utilisant les échecs réels d'eBay.

Ils discutent des résultats dans l'application des arbres de décision simplifiée sur le site d'eBay pour plusieurs mois. De plus, ils donnent une analyse coûts-avantages des systèmes de diagnostic vs le diagnostic manuel. Leurs contributions comprennent l'approche statistique de l'apprentissage, l'adaptation des arbres de décision dans le contexte du diagnostic de panne [8].

Snitch : est un prototype d'outil qui aide les techniciens en trouvant des relations entre l'Etat des applications et les défauts ultérieurs. IL fait une corrélation entre la configuration des applications et les erreurs à travers de nombreuses machines et utilisateurs et sur des longues périodes de temps. Snitch facilite à l'expert humain l'extraction des motifs d'un volume important de données par la construction d'arbres de décision qui identifient les problèmes de configuration potentiels, James Mickens et al appliquent Snitch à 114 Go de traces de configuration à partir de 151 machines sur 567 jours. Ils illustrent comment Snitch peut suggérer les erreurs de configuration dans les études de deux applications Windows : Messenger et Outlook [9].

II.2.2 Techniques basées sur Les systèmes experts

Les systèmes experts est un logiciel qui reproduit le comportement d'un expert humain accomplissant une tâche intellectuelle dans un domaine précis.

On peut souligner les points suivants :

les systèmes experts sont généralement conçus pour résoudre des problèmes de raisonnement, de classification ou de décision (diagnostic médical, prescription thérapeutique, régulation d'échanges boursiers...).

les systèmes experts sont des outils de l'intelligence artificielle, c'est-à-dire qu'on ne les utilise que lorsqu'aucune méthode algorithmique exacte n'est disponible ou praticable.

un système expert n'est concevable que pour les domaines dans lesquels il existe des experts humains. Un expert est quelqu'un qui connaît un domaine et qui est plus ou moins capable de transmettre ce qu'il sait.

Un système expert est composé de deux parties indépendantes :

– une base de connaissances elle-même composée d'une base de règles qui modélise la connaissance du domaine considéré et d'une base de faits qui contient les informations concernant le cas que l'on est en train de traiter.

– un moteur d'inférences capable de raisonner à partir des informations contenues dans la base de connaissance, de faire des déductions, etc[10].

SLCTES : Système Expert d'auto-apprentissage de Dépannage informatique :

(SLCTES) a été conçu pour indiquer le défaut, le dépannage approprié, et l'accès à la base de connaissances. Le système expert de dépannage présente pour l'utilisateur une liste de questions d'orientation, lorsqu'il répond, le système fournit des informations du problème est définit la panne. Les connaissances acquises ont été modélisées en utilisant les arbres de décision et sont représenté par des mécanismes à base de règles. Le système expert a été développé à l'aide de «si – alors » règles [11].

PimTool outils pour diagnostiquer les causes de l'échec d'un serveur HP. Narendra Dev et Bart Anderson font cela une analyse de la décharge du processeur Mémoire interne (PIM) et de maximiser l'influence de l'apprentissage de système expert d'une situation de défaillance matérielle à l'autre. L'outil est un système expert à base de règles, avec quelques règles imbriquées qui se traduisent par des arbres de décision. Les règles ont été mises en œuvre en utilisant un métalangage qui a été adapté pour le domaine de problème d'analyse de défaillance matérielle [12].

EXPERT PC TROUBLESHOOTER : Youssef Bassil propose un système expert pour diagnostiquer les problèmes informatiques et vise à automatiser le processus de maintenance et libère le technicien du dépannage manuel. Le système peut être déployé dans un environnement web, permettant à différents utilisateurs d'accéder au système sur une architecture de réseau avec des machines distribuées. Le système se compose d'une interface utilisateur, une base de règles, un moteur d'inférence, et une interface experte. En outre, le système dispose d'un module de logique floue (fuzzy-logic) [5] pour résoudre les erreurs de beep de POST, et un agent intelligent qui aide le processus d'acquisition de connaissance [13].

CFDTS Computer Fault Diagnosis and Troubleshooting System

Sylvester I. Ele et Adesola, W.A modélisent un système expert en utilisant la Méthode d'analyse et de conception d'applications orientées objet (OOAD) et UML. Ils adoptent la méthode d'établissement des faits appelé acquisition de connaissances qui est une approche fondée sur les connaissances pour extraire des faits et ils ont présenté différents ensembles de

règles dans leur système pour détecter les différents types de défaillances PC. Les données extraites des experts sont stockées dans la base de connaissances du Système. L'interface utilisateur graphique de système expert et la base de données du modèle sont développés en intégrant Netbeans, langage Java, outil de base de données MySQL et les bibliothèques de liens dynamiques (DLL) [14].

II.2.3 Techniques basées sur les Systèmes d'aide à la décision

Un élément important est le dépannage des applications misconfigured, en 2004 Helen J et al ont développé un nouveau système de dépannage : Peer Pressure, qui utilise des statistiques et l'estimation Bayésien à partir d'un ensemble d'échantillons de machines pour diagnostiquer les causes premières erreurs de configuration sur une machine malade Ils ont introduit une mesure de classement pour les ordinateur candidats de mauvaise configuration. Cette mesure est basée sur l'estimation bayésienne empirique [15].

John S. Breese et David Heckerman ont développé et étendu les méthodes existantes de la théorie de décision basée sur les réseaux bayésiens pour dépanner un dispositif qui ne fonctionne pas. Dans leur travail, ils développent une procédure de diagnostic qui sélectionne les meilleurs plans d'action en estimant le coût. Ils ont développé un système d'aide à la décision pour le dépannage où on peut choisir parmi plusieurs classes d'actions possibles de réparation d'un composant, faire une observation passive, ou de changer la configuration de système [16].

II.2.4 Techniques basées sur Le diagramme d'Ishikawa

Le diagramme d'Ishikawa ou diagramme en arêtes de poisson est un outil de diagnostic des pannes et dysfonctionnements. Il a été mis au point en 1982 par Kaoru Ishikawa pour les chantiers navals Kawasaki. Ce diagramme aide à faire l'inventaire des causes possibles d'un problème mais on peut aussi l'employer pour développer un projet Dans sa forme première, le diagramme répartit les causes en cinq catégories connues sous le nom des 5 M [17].

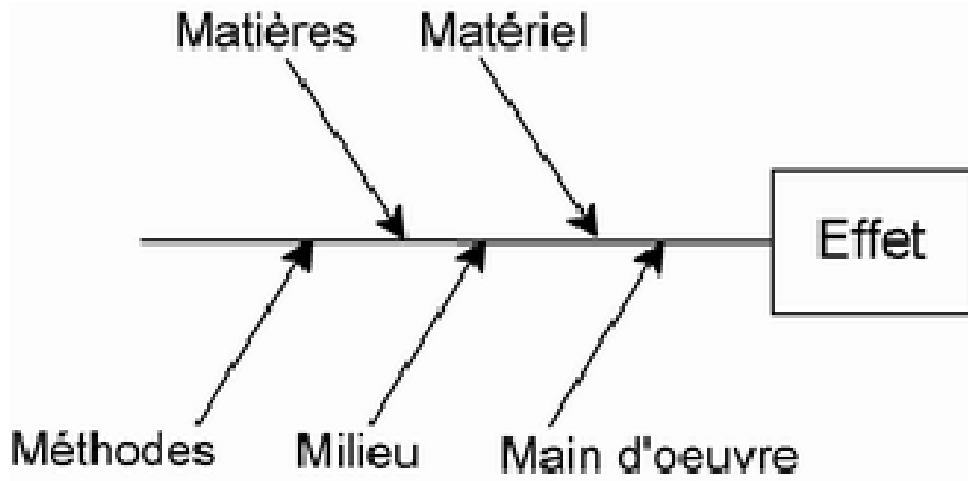


Figure II- 1 Le diagramme d'Ishikawa

H. Schyns propose un diagramme des causes découpé en deux parties (Figure II- 2) :

- les arêtes supérieures concernent le matériel.
- les arêtes inférieures concernent le logiciel.

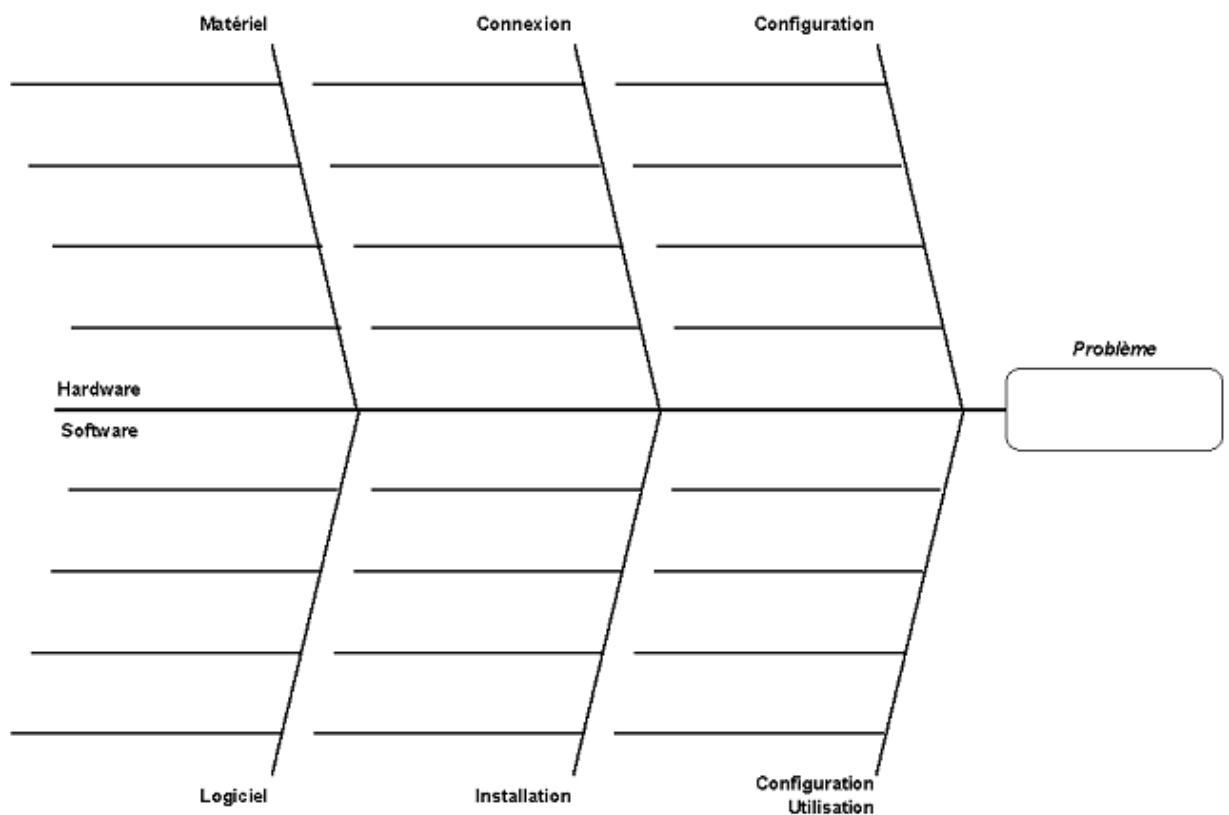


Figure II- 2 Le diagramme des causes de H.Schyns

II.2.5 Techniques basées sur les arbres des causes :

L'arbre des causes est un outil d'analyse créée en 1970 par l'Institut National de Recherche et de Sécurité sur base des travaux de la CECA (Communauté européenne du charbon et de l'acier). Il a été conçu à l'origine pour l'analyse des accidents de travail dans les mines de fer de Lorraine mais nous pouvons en faire une excellente méthode d'analyse des incidents ou problèmes informatiques [17].

L'arbre des causes est la représentation graphique de l'enchaînement logique des faits qui ont provoqué l'incident ou le problème.

L'objectif est double :

- Compréhension.
- rechercher les causes qui ont conduit à l'incident, ce qui permet de déceler des risques nouveaux ou de connaître des risques inédits ou inavoués.
- comprendre ce qui s'est passé.
- Prévention.
- éviter le retour d'un incident identique, en prenant des mesures immédiates,
- traiter les causes profondes.
- prévenir d'autres incidents, en supprimant les risques similaires dans d'autres secteurs de l'entreprise ou du système.

L'idée est que, en principe, il suffit d'éliminer l'un des faits de l'arbre des causes pour supprimer l'occurrence de l'incident.

II.2.6 Techniques basées sur des modèles :

Model-Based Design (MBD) est une méthode mathématique et visuelle de résolution des problèmes liés à la conception des systèmes complexes, le traitement du signal et des systèmes de communication... [18].

Les techniques basées sur des modèles peuvent être classés en:

1) des techniques physiques de modèles à base qui utilisent les lois physiques. Tels que les lois de la mécanique, de l'électromagnétisme, ou cinétique chimique pour modéliser le comportement du système et de déterminer si un composant anormal est présent.

2) des modèles de régression et de files d'attente qui modélisent les relations entre la consommation de ressources et le comportement de l'application. Stewart et al [19] modélisent la relation entre plusieurs ressources physiques : CPU, disque et réseau, et les temps de réponse pour un mélange de transaction.

3) des modèles de théorie des graphes qui exploitent les connaissances sur la façon dont les erreurs ou les succès se propagent dans un système pour localiser les problèmes. SCORE [20], Sherlock [21].

II.2.7 Techniques basées sur les statistiques

Les techniques statistiques pour le diagnostic résument et interprètent des données empiriques utilisant des techniques telles que la corrélation, la comparaison de l'histogramme et la théorie des probabilités [18].

Ces techniques sont centrées sur les données et nécessitent un peu de connaissances des experts ou des modèles détaillés sur les composants internes du système.

Les techniques statistiques sont soit:

1) des techniques paramétriques qui supposent que les données sont tirées par une distribution connue, par exemple, la distribution normale. Ces techniques détectent généralement un comportement anormal en identifiant les écarts significatifs de la moyenne pour les compteurs de performance, qu'ils assument suivent une distribution normale.

Toutefois, les taux de défaillance du matériel sont mieux modélisés en utilisant les distributions Weibull qui captent les taux d'échec à augmenté de dispositifs à mesure qu'ils vieillissent [22].

2) des techniques non paramétriques qui ne reposent pas sur des données appartenant à une distribution particulière, mais plutôt d'estimer la distribution à l'aide des histogrammes et estimation de la densité du noyau.

Les techniques non paramétriques estiment la distribution des données à l'aide des histogrammes et des estimateurs de densité du noyau, ou faire des généralisations sur les populations par exemple, en utilisant la corrélation.

Les techniques à base Histogramme-diagnostiquent posent généralement des problèmes en comparant les histogrammes des compteurs de performance avant et pendant une période d'anormale afin d'identifier les mesures les plus susceptibles d'être associés à ce problème. Tan et al [23].

PeerWatch [24] utilise peer-comparaison pour détecter des anomalies dans les clusters hétérogènes exécutant un matériel différent. Leur algorithme peer-comparaison utilise une technique d'analyse de corrélation canonique pour normaliser la performance des différences à cause de matériel différent, et de découvrir des corrélations entre les pairs.

II.2.8 Techniques basées sur L'apprentissage automatique :

L'apprentissage automatique ou apprentissage statistique (*machine learning* en anglais), champ d'étude de l'intelligence artificielle, concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine (au sens large) d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou impossibles à remplir par des moyens algorithmiques plus classiques[18].

Les types des techniques d'apprentissage de diagnostic qui reposent sur l'apprentissage de la machine peuvent être classés en deux grandes catégories, à savoir:

1) L'apprentissage non supervisé Il s'agit pour un logiciel de diviser un groupe hétérogène de données, en sous-groupes de manière que les données considérées comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire les données considérées comme différentes se retrouvent dans d'autres groupes distincts. Kiciman et Fox [25] utilise des grammaires hors-contexte probabilistes pour modéliser les chemins de causalité dans le système. Les règles de grammaire représentent la probabilité qu'un composant appelé une autre. Ils identifient les voies de causalité anormales en mesurant la différence entre la probabilité de la transition observée et la probabilité escomptée des transitions qui constituent le chemin causal.

2) L'apprentissage supervisé qui est une technique d'apprentissage automatique où l'on cherche à produire automatiquement des règles à partir d'une base de données d'apprentissage contenant des « exemples » (en général des cas déjà traités et validés).

Yuan et al. [26] apprendre les signatures des problèmes en analysant des séquences d'appels système. Ils ciblent les problèmes qui ont la même manifestation, par exemple, une page Web peut ne pas se charger en raison de différentes causes telles que l'adresse IP non valide ou un câble réseau débranché. Ils utilisent les SVM (Support Vector Machines) multi-classe à apprendre les signatures de problèmes.

II.3 Le ChatBot

Un « chatbot » est un robot avec lequel on peut discuter. En d'autres termes, c'est une conversation avec un partenaire virtuel. Dans le passé, les « chatbots » étaient uniquement capables d'utiliser le texte pour communiquer. Vous écriviez une question, appuyiez sur Entrée, et le système vous renvoyait une réponse à base de texte. Maintenant, avec l'arrivée des nouvelles technologies en matière de reconnaissance vocale, vous n'avez qu'à poser votre question à l'aide de votre microphone, appuyé sur Entrée, et le « chatbot » répondra à votre question comme si vous aviez une conversation avec quelqu'un sur Skype [27].

Les premiers chatterbots ont été ELIZA, PARRY, et SHRDLU.

ELIZA fut créée en 1966. Elle avait été créée par Joseph Weizenbaum, du Massachusetts Institute of Technology (MIT), et ne prenait que 3 pages en langage SNOBOL [28].

Elle commençait par poser une question neutre (Bonjour. Pourquoi venez-vous me voir ?) à la personne testée, puis analysait à chaque fois la réponse pour tenter de reposer une question en relation avec celle-ci.

Si une question lui était posée, elle demandait pourquoi on lui posait la question

Si une phrase contenait le mot computer, elle demandait : Dites-vous cela parce que je suis une machine ?

PARRY était un ancien exemple d'un chatterbot, mis en œuvre en 1972 par le psychiatre Kenneth Colby. Puis à l'Université de Stanford. PARRY a essayé de simuler un Schizophrénie paranoïde. et en tant que telle était un programme beaucoup plus sérieux et avancé qu'ELIZA [29].

SHRDLU a été l'un des premiers programmes informatiques de compréhension du langage naturel, développé par Terry Winograd au MIT à partir de 1968-1970. Il était écrit en Micro Planner et en Lisp et implémenté sur un ordinateur PDP-6.

Des ajouts ultérieurs, effectués dans les laboratoires d'infographie de l'Université d'Utah, permirent d'ajouter un rendu 3D au « monde » de SHRDLU. Par la suite, Winograd semble avoir pris ses distances avec SHRDLU et le domaine de l'intelligence artificielle, convaincu que SHRDLU constituait une impasse [30].

A.L.I.C.E. (de Artificial Linguistic Internet Computer Entity), aussi appelée Alicebot, ou tout simplement Alice, est un logiciel de robot parlant (chatterbot) un programme qui permet une conversation avec un humain en appliquant des règles heuristiques de reconnaissance de mots clé à l'entrée de l'humain. Il a été inspiré par le classique programme ELIZA de Joseph Weizenbaum. Il est l'un des plus importants programmes de ce type et a remporté trois fois le Prix Loebner relatif au Test de Turing, attribué au meilleur chatterbot en 2000, 2001 et 2004.

Alice a été développée à l'origine par Richard Wallace, elle a « vu le jour » le 23 Novembre 1995. Le programme a été réécrit en Java à partir de 1998. La mise en œuvre actuelle de la version Java est le programme D. Le programme utilise un format XML appelé AIML (Artificial Intelligence Markup Language) pour spécifier les règles heuristiques de conversation. Le code d'Alice est disponible en open source [31].

Tay est un agent conversationnel lancé le 23 mars 2016 par Microsoft sur la plateforme Twitter. Le bot a été créé par des équipes du Microsoft Technology and research et de Bing. il est nommé «Tay» après l'acronyme "thinking about you" Tay a été conçu pour imiter les modèles linguistiques d'une 19-year-old girl américaine, et d'apprendre à interagir avec des utilisateurs humains de Twitter.

Facebook a officiellement lancé son API chatbot pour Messenger, lequel il est nommé Messenger Platform. Ces chatbots sont destinés aux entreprises pour le service à la clientèle automatisé. Au cours de la conférence F8 de Facebook le 12/04/2016, Mark Zuckerberg vient d'annoncer qu'on allait bientôt être en mesure d'atteindre les entreprises à partir de la version autonome de Facebook Messenger application [32].

II.4 Conclusion

Nous avons introduit dans ce chapitre l'état de l'art qui résume les principaux travaux qui ont été menés autour d'outils et de techniques d'évaluation des pannes informatiques. Ces techniques sont principalement basées sur les arbres de décision, les systèmes experts, le diagramme d'Ishikawa, les arbres des causes, des modèles, les statistiques et les techniques basées sur l'apprentissage automatique. Nous avons aussi présenté dans ce chapitre la notion de chatbots ou agents conversationnelle.

Les chatbots sont des logiciels qui utilisent l'intelligence artificielle pour simuler des conversations et faciliter les achats en ligne.

En s'inspirant de ces différentes recherches et de la nouveauté des chatbots, notre travail sera la conception et le développement d'un outil d'évaluation des pannes informatiques que nous appellerons ExpertBot repair qui disposera de deux modules. Le premier sera basé sur la notion de systèmes experts et le second sur la notion des chatbots. La modélisation des données se fera par des arbres de décision.

Dans le chapitre qui suit, nous allons expliquer en détail l'architecture générale de ce système avec la modélisation de ces composants par le langage UML.

CHAPITRE III

LA CONCEPTION DE L'OUTIL D'EVALUATION DES PANNES INFORMATIQUES (EXPERTBOT REPAIR)

III.1 Introduction

Dans ce chapitre, nous allons présenter le processus de conception de l’outil d’évaluation des pannes informatiques. Le système ExpertBot repair est un système d’aide à la décision basée sur des données bien formulées bien structurées par des experts dans le domaine de maintenance informatique. Ce système est composé de deux modules un système expert et un ChatBot (agent conversationnel), le système expert est orienté vers des utilisateurs qui sont des débutants ne savent pas la cause de la panne. Le ChatBot est orienté vers des utilisateurs qui ont des connaissances assez importantes sur l’évaluation des pannes informatiques.

III.2 Aperçu générale du système

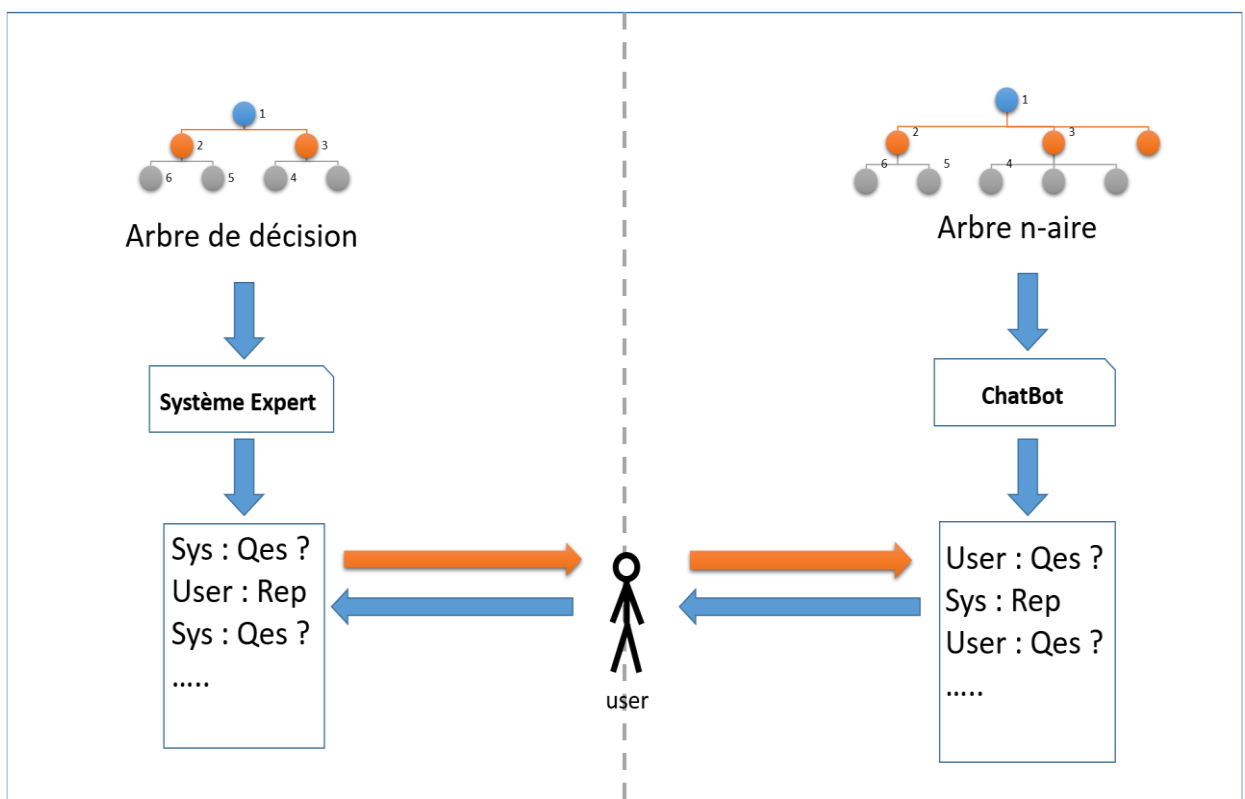


Figure III- 1L'architecture générale de systèmes ExpertBot Repair

Ce système d'aide à la décision est basé sur des données et informations bien préparées par des experts et donc nous avons présenté ces données sous forme d'arbres de décision. Le système se compose de deux modules : un système expert et un agent conversationnel(ChatBot).

Les données des systèmes experts sont modélisées sous la forme d'un arbre de décision. Qui représente une hiérarchie de tests et de classification des pannes informatiques. Les arbres n_aires sont utilisés pour représenter la base de connaissance du système de chat parce que les données de chat son très larges et sont des données très détaillées.

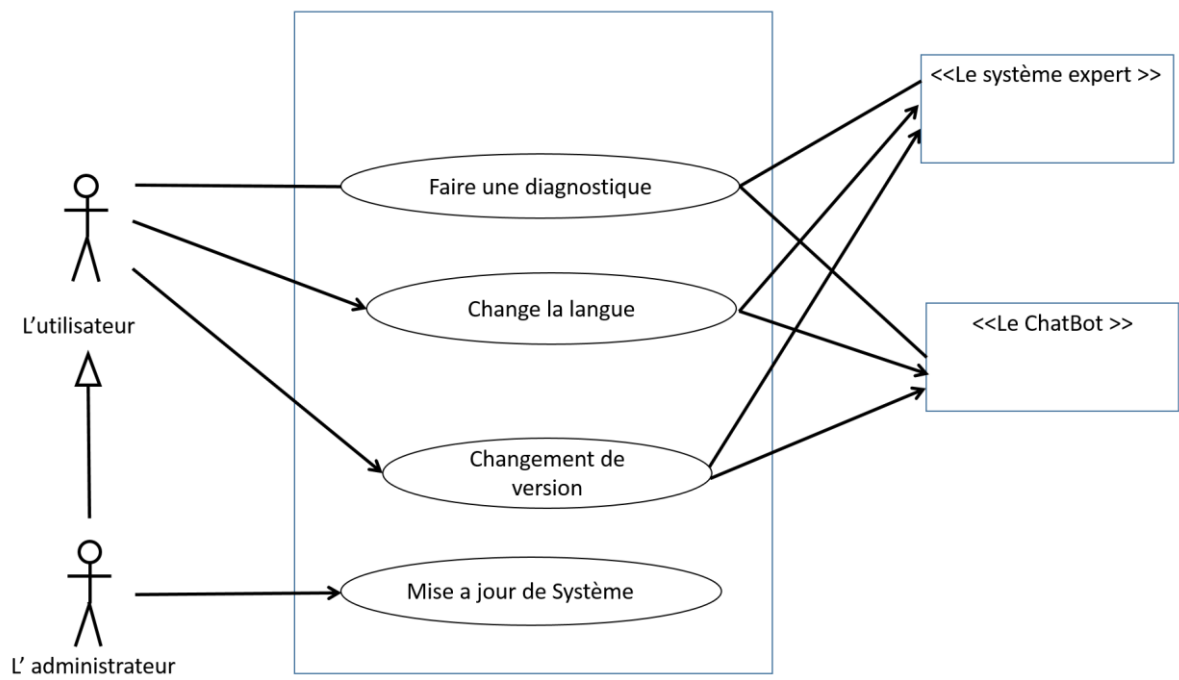
Les arbres n-aires qui sont une généralisation des arbres binaires : chaque nœud a au plus n fils. On peut utiliser un arbre n-aire pour Les données hiérarchiques et pour accélérer la recherche.

Le rôle du système expert est de questionner l'utilisateur pour le guider vers la cause de la panne et enfin lui donner la solution selon la chaine des réponses. Le but des questions est de connaître l'état des composants matériels avec des réponses de l'utilisateur qui sont des « oui ou non », donc une question après l'autre le système va connaître la cause .Le système expert est orienté vers des utilisateurs débutants en informatique.

Le ChatBot par contre est un module orienté vers des utilisateurs qui ont des connaissances sur les pannes informatiques ou qui connaissent déjà la cause de la panne. Dans ce cas l'utilisateur va communiquer avec le système soit par des phrases ou des questions et le système lui répond par des solutions ou des informations.

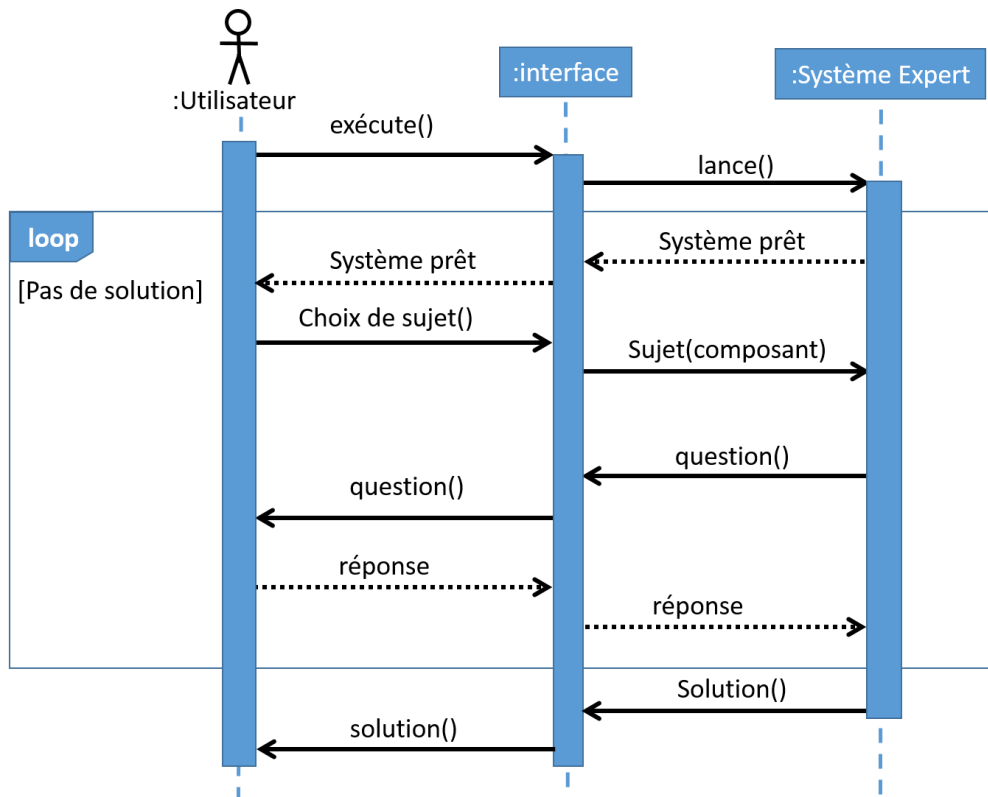
III.3 La conception de l'outil

III.4.1 Le diagramme de cas d'utilisation

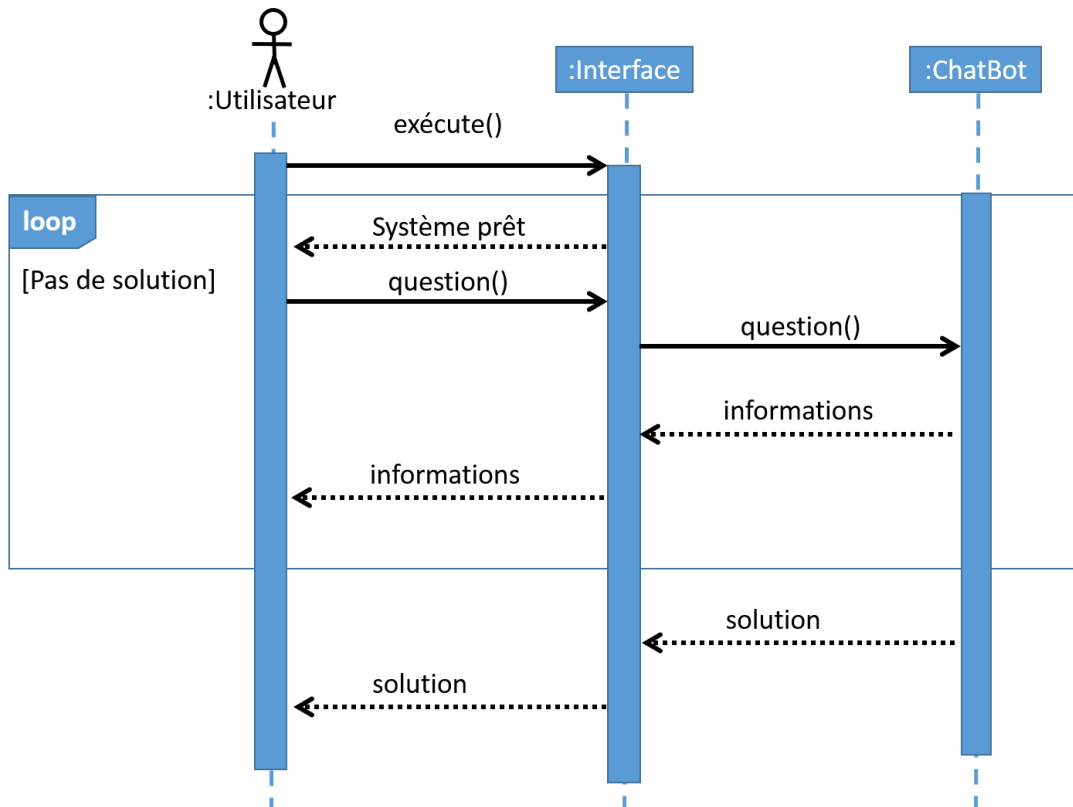


III.4.2 Les diagrammes de séquence

1) Le diagramme de séquence diagnostique par système expert



2) Le diagramme de séquence diagnostique par ChatBot



L'identification

Nom : faire un diagnostic (package « System Expert » ou package « ChatBot »).

Acteur(s) : système expert ou chatbot ou l'utilisateur.

Description : faire une communication entre le système et l'utilisateur pour faire un diagnostic et évaluer la panne.

Pré-conditions : L'utilisateur doit choisir un sujet pour le discuter avec le système.

Démarrage : L'utilisateur lance le système expert ou le chatbot.

Description des scénarios

Le scénario nominal

1. Le système affiche une interface contenant la liste des catégories de pannes et composant (pannes de démarrage, pannes d'alimentation, pannes audio...).
2. L'utilisateur sélectionne une catégorie.
3. Le système affiche une question pour l'utilisateur afin de choisir le mode d'évaluation
4. L'utilisateur choisit un mode d'évaluation (système expert ou chatbot).
5. Le mode ce qui l'utilisateur a le choix de lancer le diagnostique.
6. L'utilisateur répond aux questions de système expert ou donne l'état des composants au chatbot.
7. Le système donne la cause de la panne et la solution.

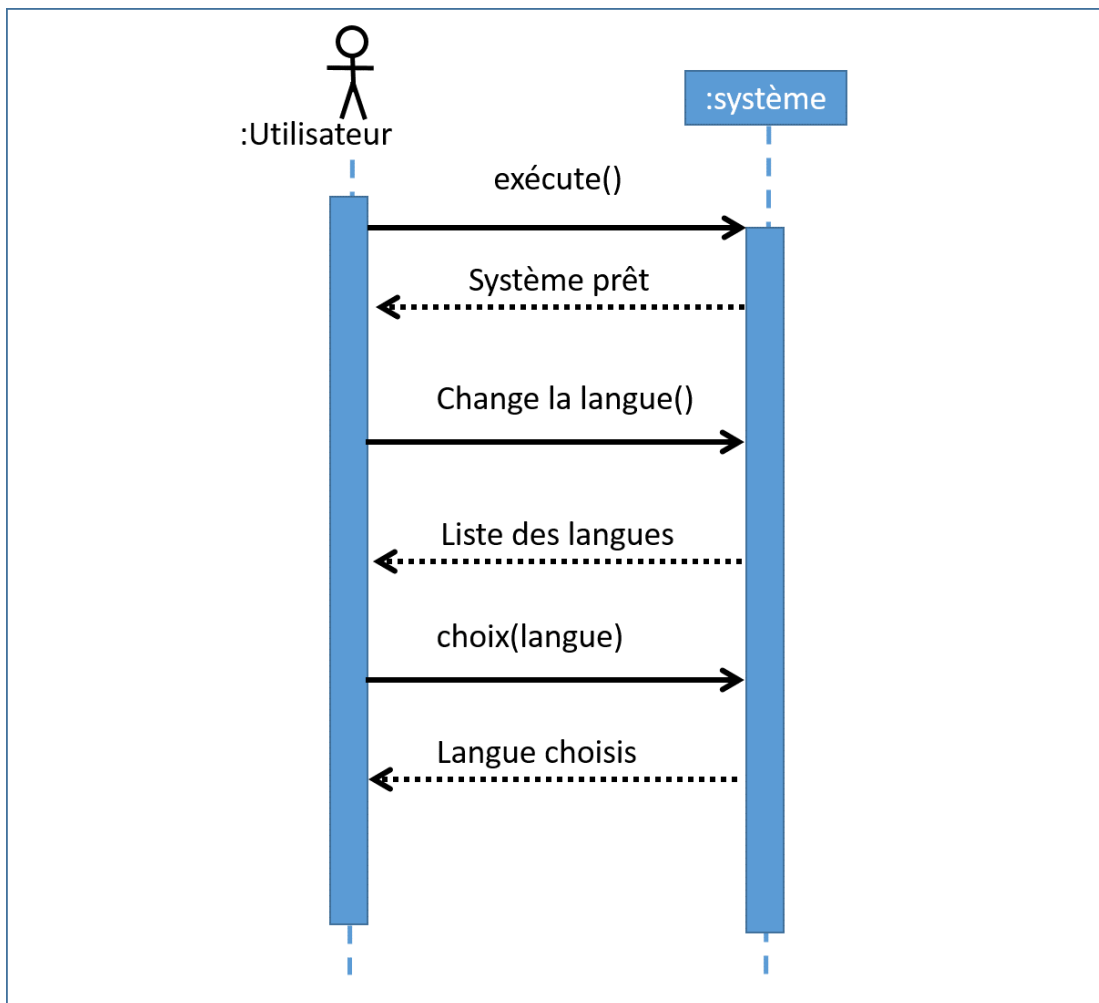
Les scénarios alternatifs

- 2.a l'utilisateur peut quitter le système.
- 4.a l'utilisateur choisir le mode système expert.
- 4.b l'utilisateur choisit le mode chatbot.
- 4.c l'utilisateur peut faire le retour au menu principale.

Fin : Scénario nominal : aux étapes 2 ou 4, sur la décision de l'utilisateur.

Post-conditions : Aucun.

3) Le diagramme de séquence de Change la langue



L'identification

Nom : Change la langue (package « System Expert » ou package « ChatBot »).

Acteur(s) : L'utilisateur.

Description : L'utilisateur change la langue du système pour une bonne communication.

Pré-conditions : L'utilisateur lance le système expert ou le chatbot.

Démarrage : L'utilisateur choisit la langue à partir du menu langue.

Description des scénarios

Le scénario nominal

1. Le système affiche l'interface principale.
2. L'utilisateur click sur le bouton langue.
3. Le système affiche la liste des langues.
4. L'utilisateur choisit une langue à partir de la liste.
5. Le système redémarre.

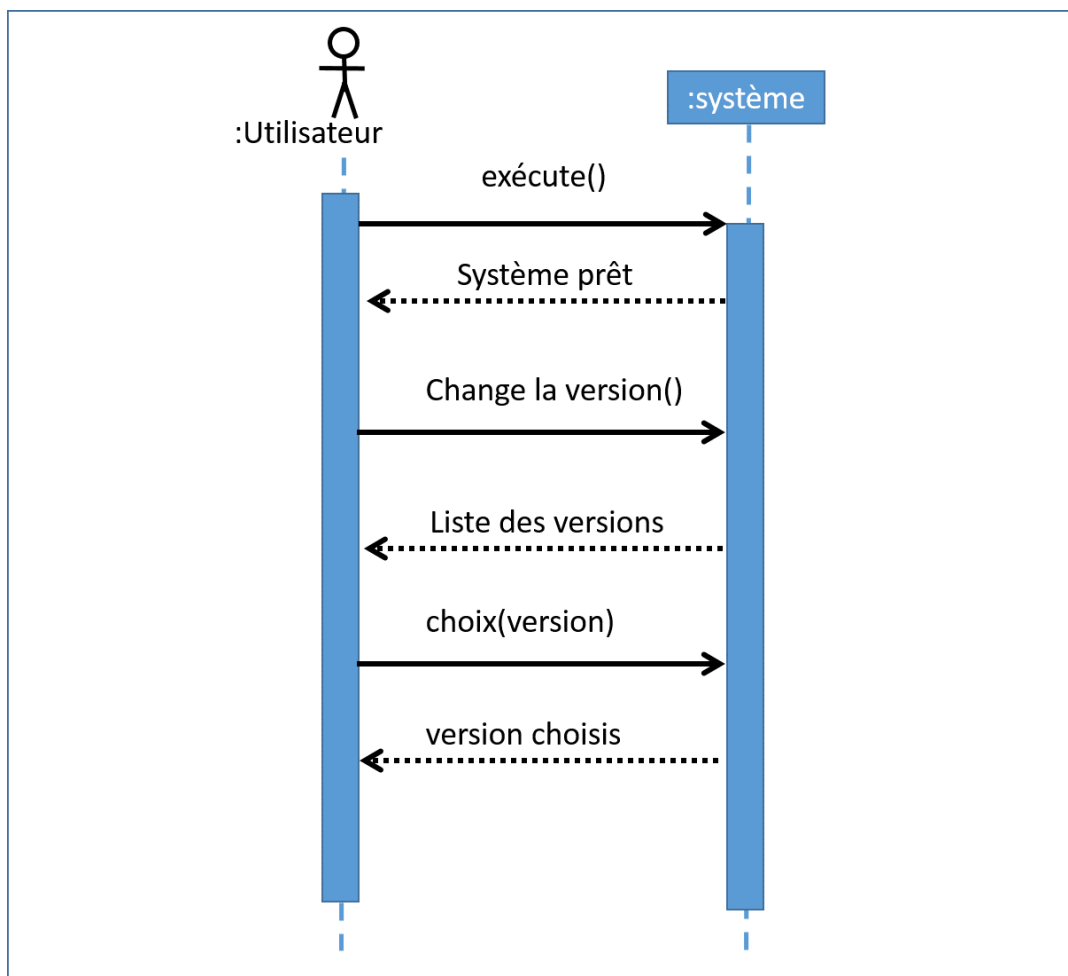
Les scénarios alternatifs

2.a l'utilisateur peut quitter le système.

Fin : Scénario nominal : à l'étape 1 sur la décision de l'utilisateur.

Post-conditions : Aucun.

4) Le diagramme de séquence de Changement de version



L'identification

Nom : Changement de version (package « System Expert » ou package « ChatBot »).

Acteur(s) : l'utilisateur.

Description : l'utilisateur demande une nouvelle version du système.

Pré-conditions : L'utilisateur lance le système et fait un click sur le bouton update.

Description des scénarios

Le scénario nominal

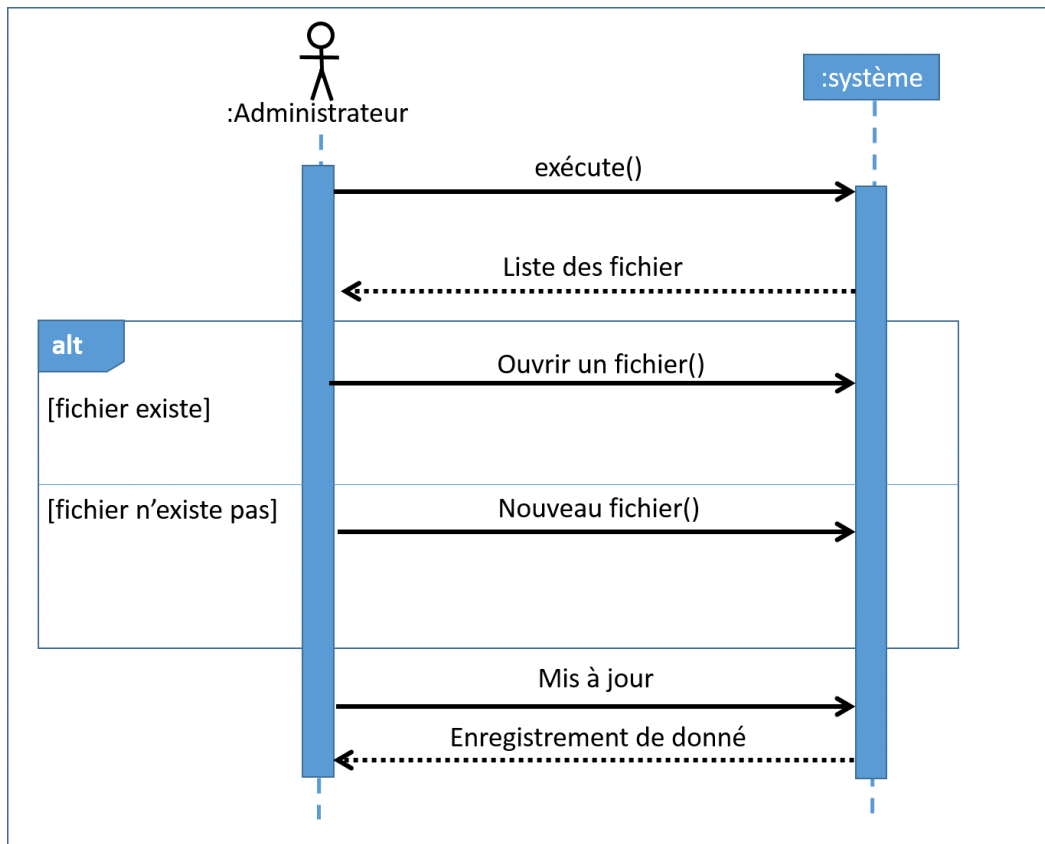
1. Le système affiche une page contenant la liste des composants.
2. L'utilisateur click sur le bouton update.
3. Le système affiche une fenêtre pour gérer l'opération de mise à niveau.
4. L'utilisateur confirme la demande de mise à niveau avec la version.
5. Le système fait la mise à niveau.

Les scénarios alternatifs

- 2.a l'utilisateur peut quitter le système.
- 2.b l'utilisateur peut quitter le mise à niveau.

Fin : Scénario nominal : aux étapes 2 ou 4, sur la décision de l'utilisateur.

5) Le diagramme de séquence de Mise à jour de Système



L'identification

Nom : Mise à jour de Système (package « System Expert » ou package «ChatBot »).

Acteur(s) : l'administrateur.

Description : l'administrateur enrichi les bases de connaissance, et recherche et corrige les éventuels bugs du système.

Pré-conditions : l'administrateur lance l'application admin.

Description des scénarios

Le scénario nominal

1. L'administrateur lance l'application admin.
2. L'administrateur peut ouvrir un fichier de base de connaissance de système expert, chatbot ou créer un nouveau fichier.

III.4 Le développement du système expert

III.4.1 Le système expert

Nous allons décrire les principes de base de ce système expert. Tous les systèmes experts comprennent au moins trois éléments de base: La base de faits, le moteur d'inférence et enfin une interface avec l'utilisateur.

La base de faits, qui contient les connaissances nécessaires à la pratique et les informations des composants d'ordinateur, le moteur d'inférence partie fabriquant des raisonnements sous forme des conditions pour une bonne définition de problème. L'interface pour faciliter la communication entre l'homme et la machine.

On a utilisé CLIPS pour construire le système expert L'origine de CLIPS (CLIPS 1.0) est une version « C » du système « ART » (Automated Reasoning Tool) réalisé par la NASA et distribué « gratuitement » (jusqu'à version 5.0). CLIPS est basés sur l'algorithme RETE. Depuis, CLIPS a évolué et inclut la programmation orientée objet et un système de maintenance. CLIPS est un système ouvert. Il est distribué avec sa source. Il est facile d'ajouter les procédures écrites en C. CLIPS est le noyau pour plusieurs milliers de systèmes expert [33].

III.4.2 La base de faits

Elle est l'élément capitale, car elle contient la représentation des connaissances de l'expert Lorsque, au sein d'un groupe de connaissances, les concepts entretiennent entre eux des relations hiérarchiques, on utilise un arbre de décision binaire. La formalisation des savoirs se fait très souvent sous forme de règles de production, de propositions du type « si..., alors... ». Concernant notre outil, l'arbre de décision ci-dessous représente la modélisation de la connaissance concernant les pannes informatiques.

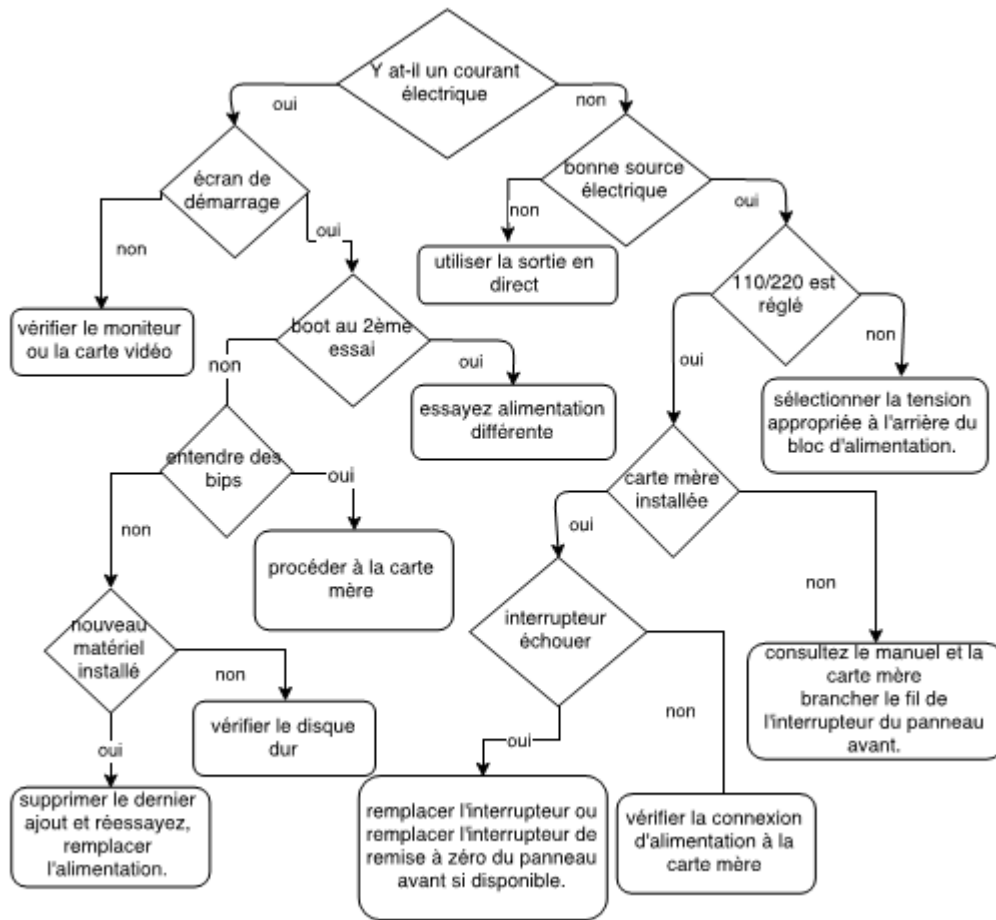


Figure III- 2L'arbre de décision source [34]

Un système expert est une machine à états fini. Ces états sont les "contextes" (ou phases). Ces contextes peuvent être organisés en cycles, en réseaux ou en arbres. Les transitions peuvent être codées par règles (procédurales) ou par faits (déclaratifs). Le contrôle peut être autre chose qu'une séquence de phases.

Les arbres de décision sont utiles pour les problèmes de classification et les caractéristiques d'un problème de classification sont : L'ensemble des réponses possibles est fini et connu d'avance. Exemples : problème de diagnostics et de taxonomie. L'espace de solutions est réduit par une série de tests. Un arbre de décisions est composé de nœuds et de branches. Les nœuds peuvent être les décisions ou les pannes ou symptômes. Les ensembles de réponses possibles sont les feuilles de l'arbre.

III.4.3 Description de la base de connaissance :

Cette description décrit les nœuds de l'arbre de décision en chaque exécution de l'application. Ces données sont enregistrées dans la mémoire vive, dans chaque nœud identifié par un numéro comme clé et les données présentées au-dessus.

Tableau 2 la représentation d'un nœud d'arbre de décision

Identificateur	génééré automatiquement par le système CLIPS
affichage	affiché les données à L'utilisateur
Nom de relation	identifier le nom de nœud
Solution	Identifié est ce que la solution est trouvé ou non
réponse	Indique que le système trouve la solution ou non
état	Indique l'état de question

III.4.4 Description des règles

```
(defrule nom de la règle
(logical (nom de relation oui/non))
=>
(assert (UI-state (affichage PowerOnQuestion)
(nom de relation power-starts)
(solution oui/non)
(réponse oui/non))))
```

La description des règles est modélisé sous la forme des si (condition) alors (opération) on va utiliser le terme « **defrule** » pour définir une nouvelle règle avec un nom. On utilise « **logical** » pour faire un test de choix de l'utilisateur. Donc si la condition est satisfaite le système va sauvegarder les données de l'affichage au fur et à mesure pour donner à la fin une solution possible.

III.4.5 Le moteur d'inférence

Il élabore la solution en choisissant les règles de production et leur séquence d'utilisation. Quand le système met en jeu des procédures logiques classique, les raisonnements peuvent se fonder sur des faits connus (on parle de "chaînages avant"). Le moteur est constitué par des règles qui nous permettent de naviguer dans l'arbre jusqu'à trouver une solution. Chaque règle va contrôler un état dans l'arbre soit initial ou final ou intermédiaire [35].

III.4.6 Exemple

```
(defrule retour_en_ariere  
?Variable1<- le id courent  
?Variable2<- contient 2 id le id courent et le id précédent  
  
=>  
  
(retract ?Variable1) supprime dans la mémoire le contenu de variable 1 (supprimé le id courent)  
  
(modify ?Variable2 (courent ?id))  
  
Modifié le variable2 : changement de contenu de id courant par le id précédent  
  
(halt)) arrêté l'operation
```

III.5 Le développement du chatbot

III.5.1 Présentation

Un agent conversationnel ou ChatBot est un agent qui dialogue avec un utilisateur via une conversation parlée ou même une conversation écrite. Le ChatBot peut fonctionner sur les ordinateurs ou les téléphones, bien que la plupart du temps accessible par l'Internet. Donc notre système ChatBot est mobilisé ci-dessous.

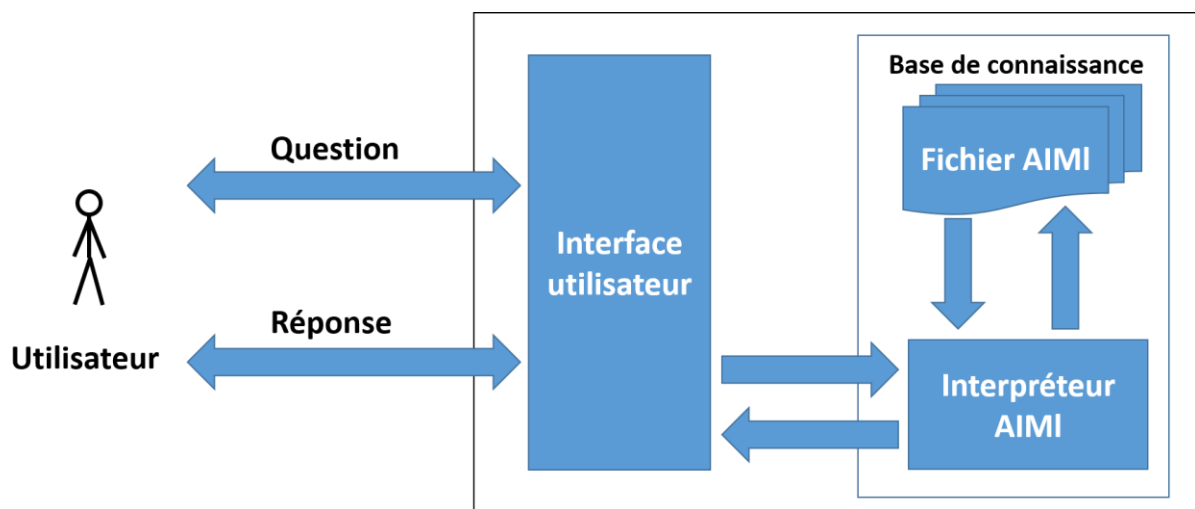


Figure III- 3Le schéma illustratif de système ChatBot

L'interface utilisateur pour la communication textuelle et vocale avec le système composé en un espace d'affichage de conversation. Un champ de texte pour écrire des messages au système et un bouton pour l'envoi des messages au système, La base de connaissance de système ChatBot est un arbre n-aire. Certains auteurs préconisent pour ce type d'arbre une implémentation par des langages de balisage .comme XML Nous utilisons le langage AIML (**Artificial Intelligence Markup Language**) sous formes de fichiers contenant les données et les connaissances utiles pour réussir une conversation.

Logiquement il faut que le système dispose d'un lien entre l'interface écrit en java et les fichiers AIML, pour cela on a utilisé l'interpréteur « ProgrammeAB ». « ProgrammeAB » sert à la mise en œuvre de référence d'AIML 2.0.

La base de connaissance du chatbot est modélisée par l'arbre binaire ci-dessous. Cet arbre est élaborée à partir des données retrouvées dans le livre « Upgrading and repairing pcs. Scott Mueller ».

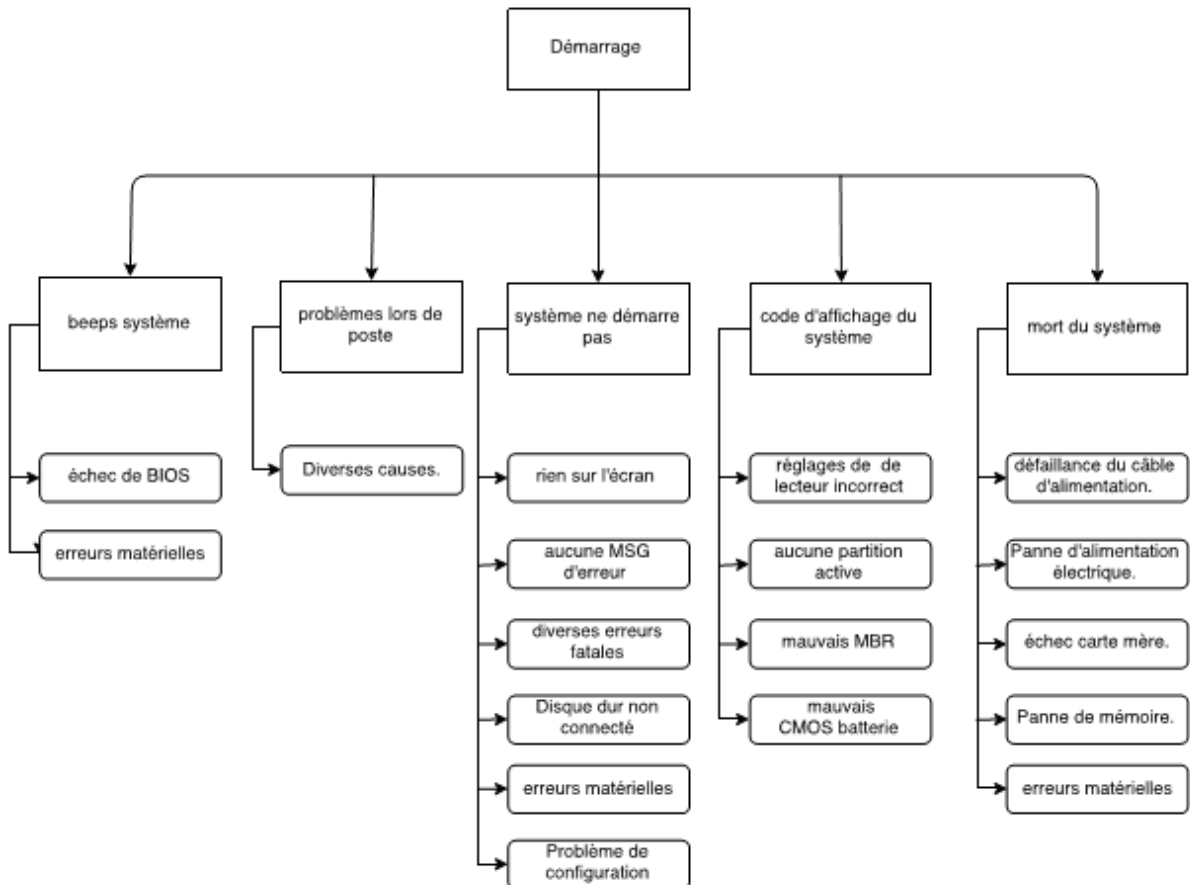


Figure III- 4 L'arbre N-aire

III.5.2 L'Artificial Intelligence Markup Language

AIML est un langage dérivé de XML utilisé pour gérer la connaissance bots (robots virtuels). Ce langage utilise une vingtaine de balises de base. Le langage XML appelé AIML a été développé par Richard Wallace et par la communauté mondiale du logiciel libre entre les années 1995 et 2002. Il a formé la base pour ce qui était initialement une extension d'Eliza appelée « A.L.I.C.E. » (Artificial Linguistic Internet Computer Entity), qui a remporté le concours annuel du Prix Loebner d'Intelligence Artificielle¹ à trois reprises, et a également été le champion du Chatterbox Challenge² (Concours de Chatterbox) en 2004 [36].

AIML contient plusieurs éléments. Les plus importants sont décrits plus en détail ci-dessous.

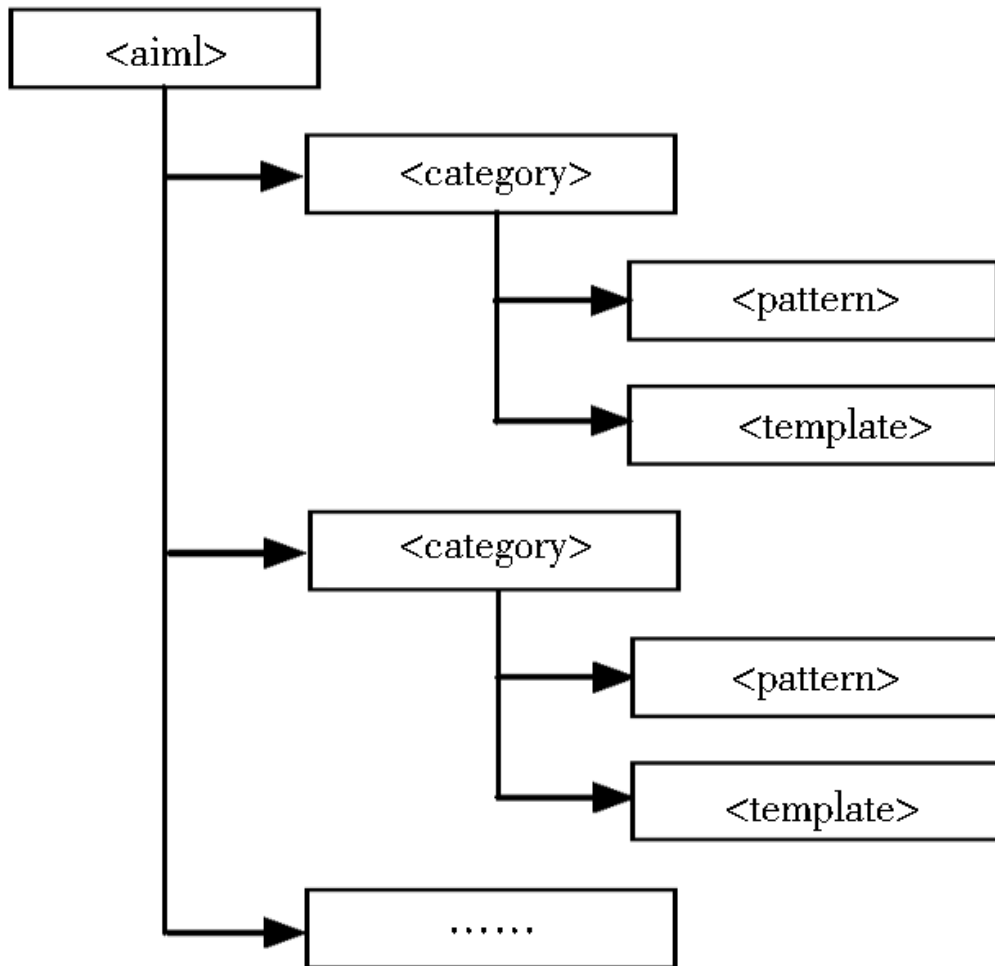


Figure III- 5La structure du langage AIML

Le principe de fonctionnement d'un chatbot est simple: l'utilisateur soumet une question ou une phrase, puis le moteur du programme va détecter le(s) mot(s)-clé correspondant dans sa base de données. Le tout est organisé à l'aide de documents AIML contenant des cas de figures de discussions et les comportements à adopter. Les données entrées par l'utilisateur sont entre les balises <pattern> et le système doit répondre par les données stockées entre les balises <template> et donc chaque question et réponse va être enregistrée entre les balises « category »

III.5.3 Exemple

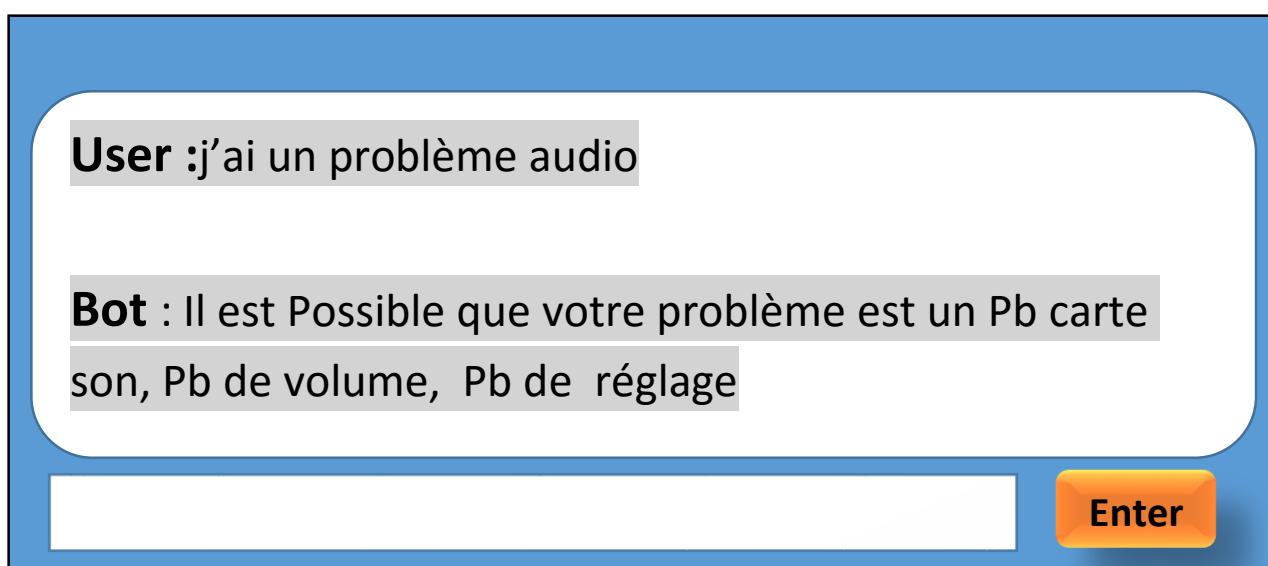


Figure III- 6Exemple d'une application de ChatBot

Dans cette exemple toute donnée écrite par l'utilisateur est ignorée juste le mot clé audio est reconnu par le chatbot et donc le système donne sa réponse par rapport à ce mot clé .La modélisation par AIML de cet exemple sera sous la forme suivante :

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <aiml>
4
5      <category>
6          <pattern> _ audio *</pattern>
7
8          <template>
9              possible votre problème est un
10             Pb carte son, Pb devolume, Pb de réglage
11
12          </template>
13
14      </category>
15
16 </aiml>
17
```

Figure III- 7Le fichier source AIML de l'exemple

III.6 Conclusion

Dans ce chapitre, nous avons présenté notre système « ExpertBot repair » qui est un système d'évaluation des pannes informatiques. Nous avons présenté l'architecture générale de cette plateforme de diagnostic avec le type de base de connaissance de chaque module. Nous avons utilisé un langage de modélisation unifié UML pour la conception de composant de systèmes.

Dans le chapitre suivant nous allons présenter l'application que nous avons développée d'une manière adéquate à la conception proposée dans ce chapitre.

CHAPITRE IV : L'IMPLEMENTATION

IV.1 Introduction

La phase d'implémentation est le résultat de la modélisation. Pour cela, ce quatrième chapitre est consacré à l'illustration de notre système d'évaluation des pannes informatiques « ExpertBot repair ». Notre but est de développer une application simple à utiliser.

IV.2 II. Plateformes logicielles

Nous avons implémenté notre système en JAVA FX, qui est une technologie créée par Sun Microsystems qui appartient désormais à Oracle, à la suite du rachat de Sun Microsystems par Oracle le 20 avril 2009.

Avec l'apparition de Java 8 en mars 2014, JavaFX devient l'outil de création d'interface graphique (GUI toolkit) officiel du langage Java, pour toutes les sortes d'application (applications mobiles, applications sur poste de travail, applications Web...), le développement de son prédécesseur Swing étant abandonné.

JavaFX est désormais une pure API Java (le langage de script spécifique qui a été un temps associé à JavaFX est maintenant abandonné). JavaFX contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc [37].

Nous avons utilisé la version JDK1.8 dans un ordinateur portable doté d'un processeur Intel core I3 @ 2.10 GHz (2 CPUs), avec une mémoire vive (RAM) de 6GO qui fonctionne sur le système d'exploitation Microsoft Windows 10 Professional 64 bits. Pour l'éditeur, nous avons choisi L'IDE (Integrated Development Environment) NetBeans IDE 8.0.2. Il s'agit de la dernière version de NetBeans, ce dernier est parmi les éditeurs java les plus appréciés, cette forte appréciation est dû à de multiples avantages notamment les simplifications d'édition et la facilité de la création des interfaces graphiques par l'option drag and drop. Ils intègrent les fonctionnalités suivantes :

1. Editeur de textes avec coloriage syntaxique.
2. L'option auto Complète (menus contextuels suggère de multiples choix).
3. Génération automatique des conteneurs et dossiers nécessaires à l'organisation d'un projet et des paquetages des classes.
4. Intégration des commandes Java et de leurs options dans des menus et des boîtes de dialogue appropriés.

5. Débogueur pour corriger les erreurs.

6. Proposition pour la résolution automatique de quelques erreurs de programmation.

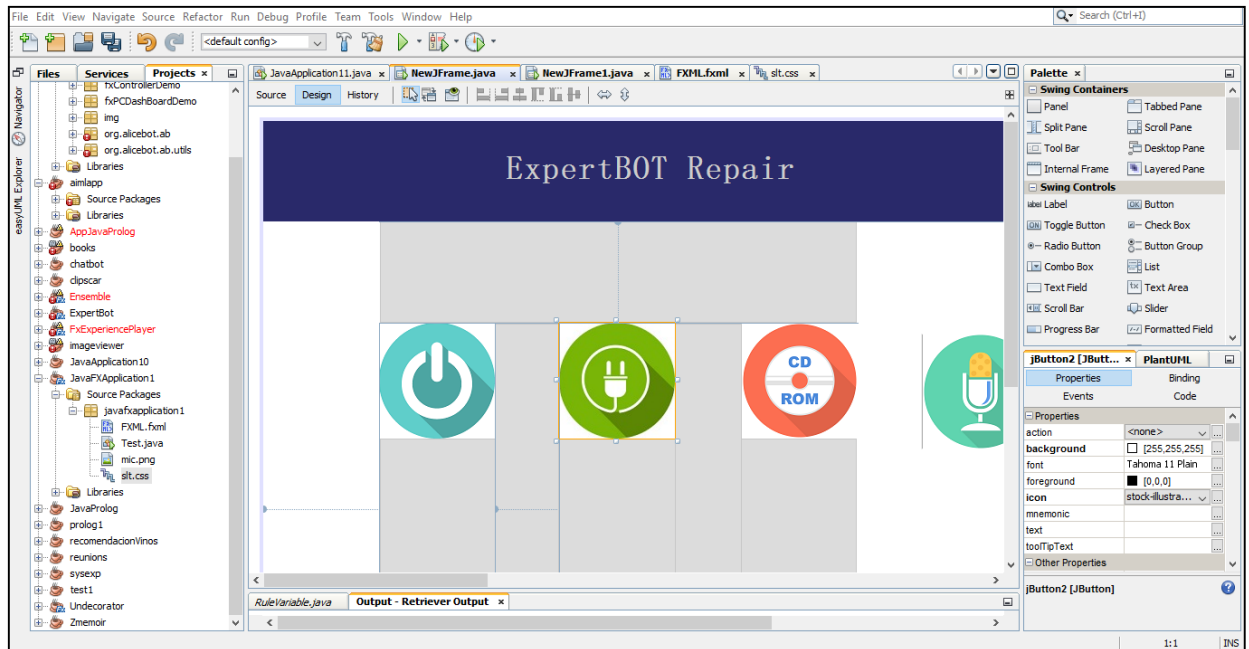


Figure IV- 1 Le Netbeans IDE

IV.3 Les interfaces

Dans le chapitre précédent nous avons vu l'architecture générale de l'outil avec les deux modes d'évaluation le système expert et l'agent conversationnelle. Nous allons présenter l'implémentation de notre système dans cette partie.

IV.3.1 Le système Expert Bot Repair :

Le menu principal est composé de 12 boutons. Chaque bouton représente un composant dans le micro-ordinateur d'autre part le bouton « Startup » représente les pannes de démarrage. Le bouton « internet » et « système » sont de type réseau et logicielle. Les options « Update » pour demander une nouvelle version et « Language » pour changer la langue de système.

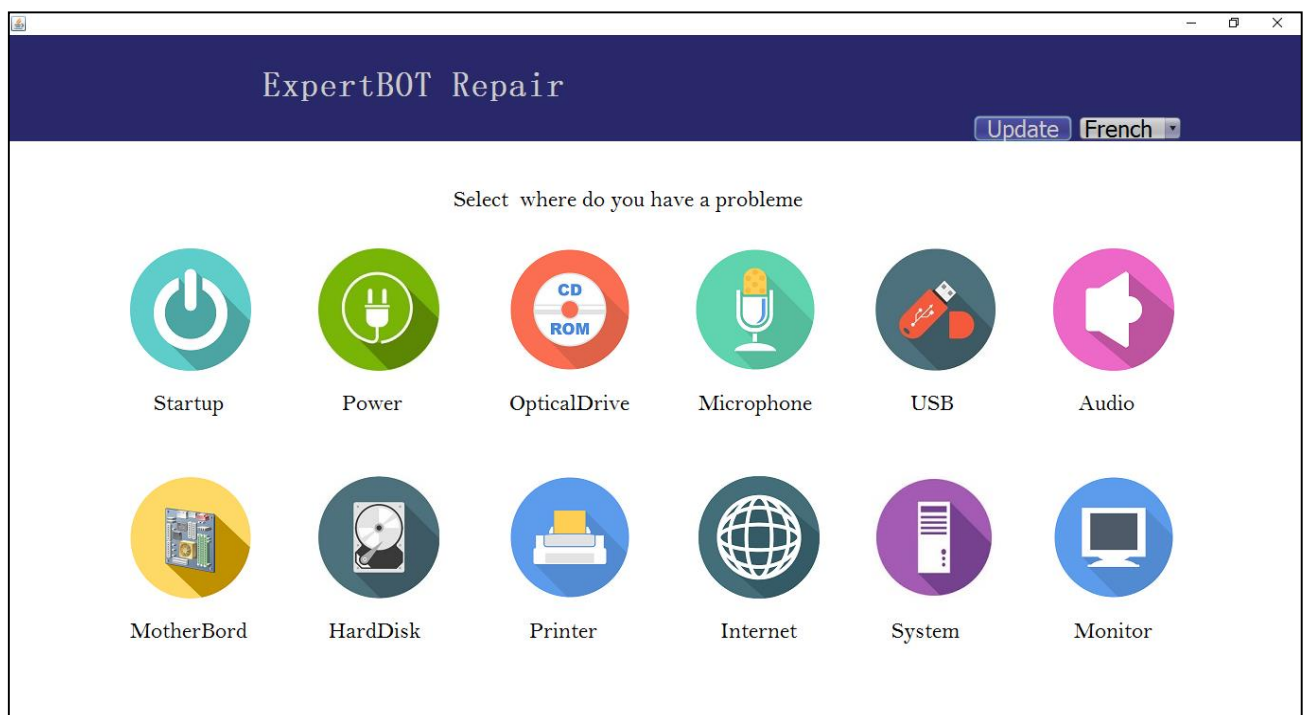


Figure IV- 2 L'Interface graphique de système ExpertBot Repair.

IV.3.2 La phase de choix de mode d'évaluation :

Après le click sur un bouton dans le menu principal une fenêtre s'affiche pour choisir le mode d'évaluation de panne. La sélection de mode va orienter l'utilisateur vers le mode préféré.

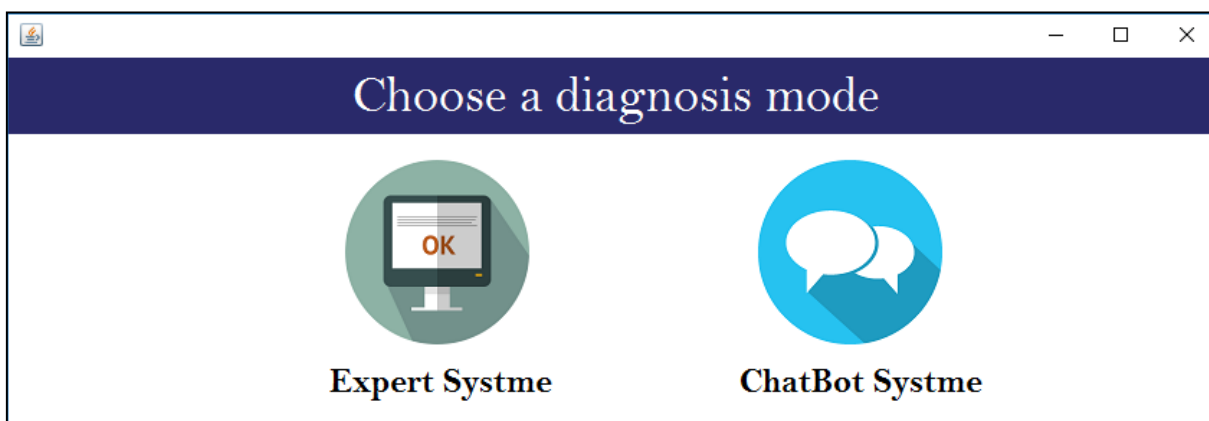


Figure IV- 3 La fenêtre de choix de système d'évaluation

IV.3.3 Le mode système expert

Le système expert est le mode où le système questionne l'utilisateur. L'utilisateur répond à ces questions par oui ou non et passe à la question suivante par le bouton « Next » Jusqu'à ce que le système expert détecte le problème et donne la solution. L'utilisateur par un click sur le bouton « Previews » peut retourner à la question précédente. Le bouton « Restart » est pour répéter le diagnostic.

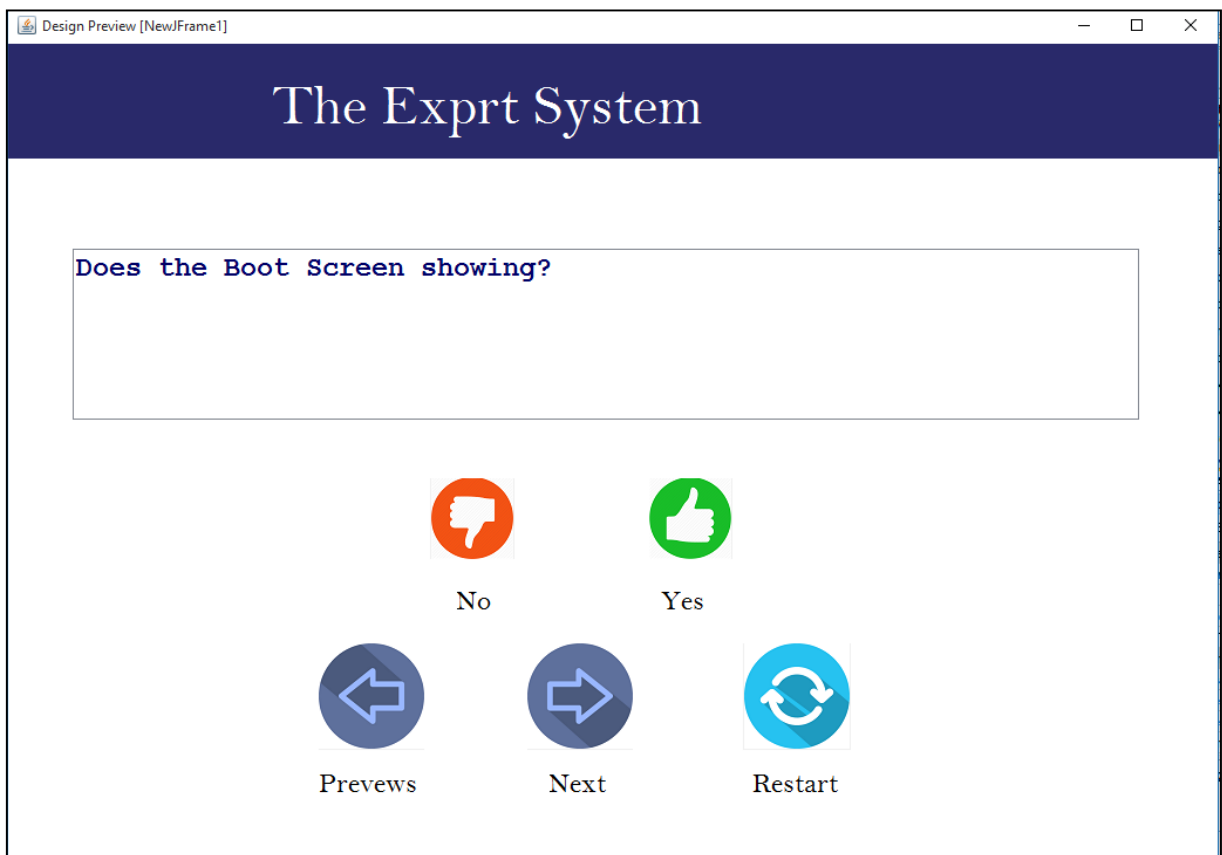


Figure IV- 4 L'interface de système expert

IV.3.4 Le mode ChatBot

Ce mode est pour discuter avec le système les symptômes et les états des composants. L'utilisateur écrit ces questions dans le champ de texte et click sur le bouton « Enter » pour faire passer la question au système.

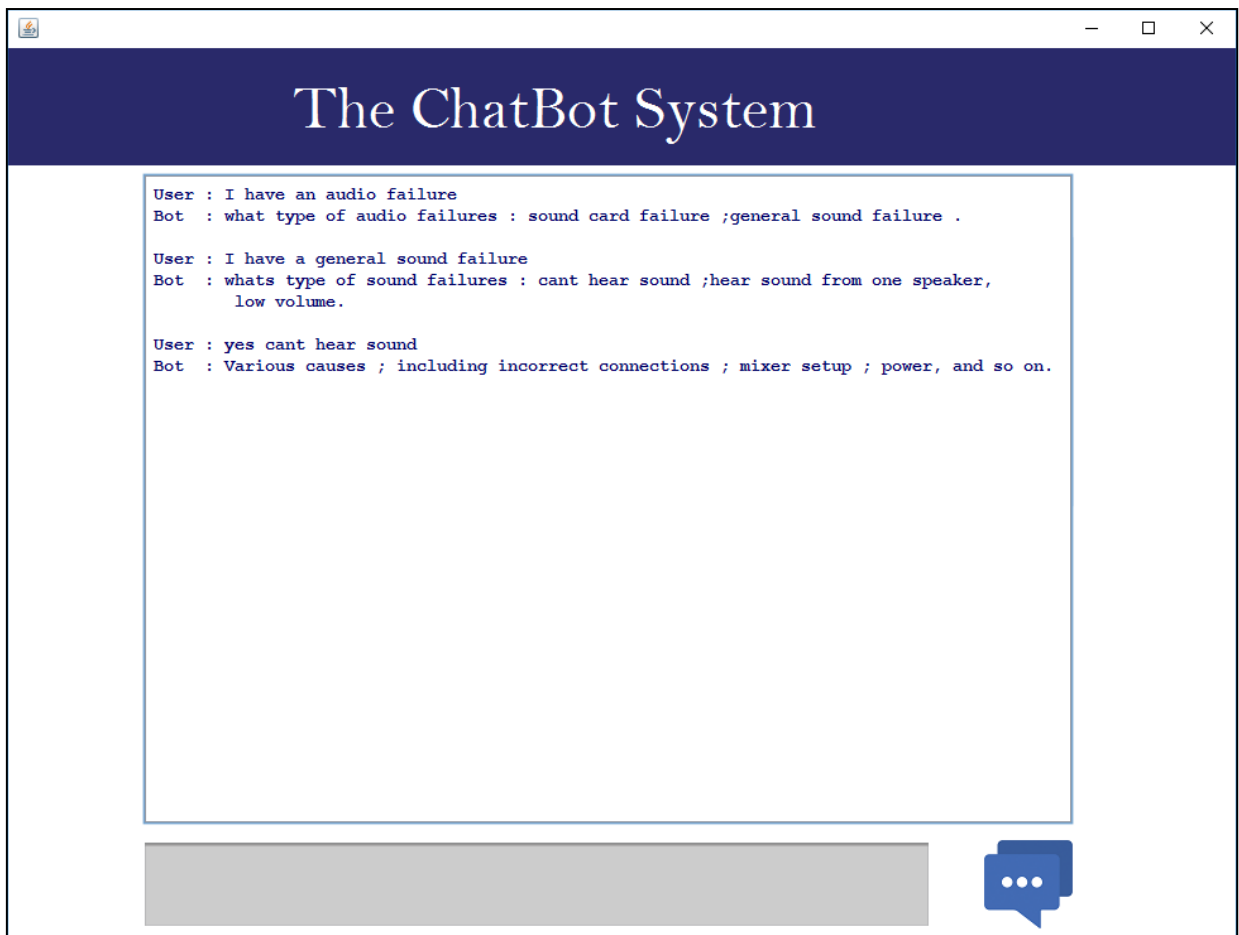


Figure IV- 5 L'interface de système ChatBot

IV.3.5 L'application d'administrateurs :

Est une application d'édition de texte orienté vers la modification du contenu de bases de connaissance du système. L'administrateur peut créer un nouveau fichier ou ouvrir un fichier existant. Après la modification l'administrateur peut sauvegarder son travail.

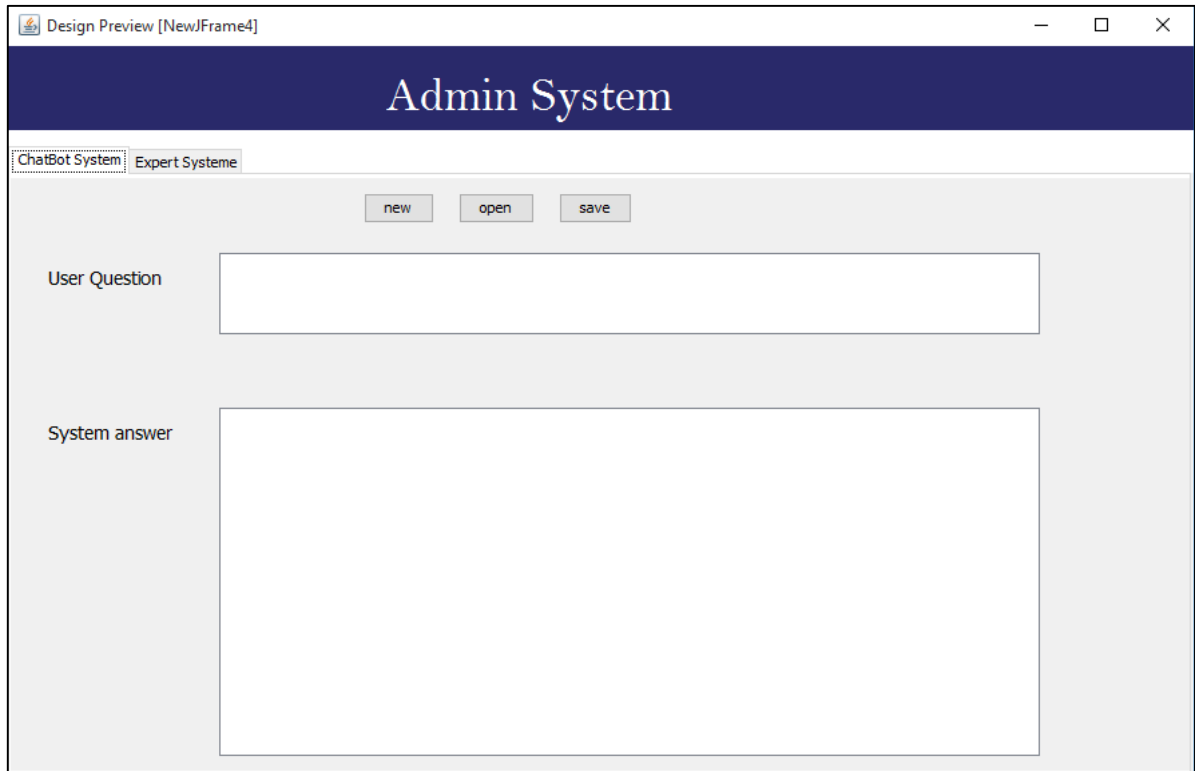


Figure IV- 6 L'interface de système d'administrateur

IV.4 Structure des programmes :

IV.4.1 Les codes des fonctions principales de l'application

Cette fonction permet d'établir la connexion entre l'interface graphique de JavaFx et les fichiers de base de connaissance AIML.

```
public void start(Stage primaryStage)    {

    MagicStrings.setRootPath();
    AIMLProcessor.extension = new PCAIMLProcessorExtension();
    MagicBooleans.jp_tokenize = false;
    MagicBooleans.trace_mode = true;
    if (MagicBooleans.trace_mode) System.out.println("Working
Directory = " + MagicStrings.root_path);
    Graphmaster.enableShortCuts = true;
    bot = new Bot(bantam, MagicStrings.root_path, action); //
    if (MagicBooleans.make_verbs_sets_maps)
Verbs.makeVerbSetsMaps(bot);
    if (bot.brain.getCategories().size()
<MagicNumbers.brain_print_size) bot.brain.printgraph();
    if (MagicBooleans.trace_mode) System.out.println("Action =
'" + action + "'");
    if (action.equals("chat") || action.equals("chat-app")) {
        booleandoWrites = ! action.equals("chat-app");
        chatSession = new Chat(bot, doWrites);
        bot.brain.nodeStats();
    }
}
```

Cette partie de code fait l'opération de lecture de texte écrit par l'utilisateur. Après le clic sur le bouton « Entrer » la fonction « GetAnswer » va importer la réponse mémorisée dans la base de connaissance et elle affichera à l'utilisateur cette réponse.

```
    Button btn = new Button("Enter");
btn.setOnAction(new EventHandler<ActionEvent> () {
    @Override public void handle(ActionEvent e) {
        textPane.appendText("user :"+txtInput.getText()+"\n");
        String rep= GetAnswer() ;
        textPane.appendText("bot  :"+rep+"\n");
    }
});

    Scene = new Scene(grid, 1000, 875);
    scene.getStylesheets().add("/application/Styl.css");
primaryStage.setScene(scene);
    primaryStage.setMaximized(true);
    primaryStage.show();
}

public String GetAnswer(){
    String question=null;String reply =null;
    question = txtInput.getText().toString() ;
    System.out.println(question);
    reply = TestAB.testChatBot(chatSession, bot, question);
    System.out.println(reply);

    return reply;
}
}
```

Cette partie de code contient la fonction de navigation dans l'arbre de décision par les boutons « Next » « Preview » et « Restart ».

```
if (ae.getActionCommand().equals("Next"))
{
    if (choicesButtons.getButtonCount() == 0)
    { clips.assertString("(next " + currentID + ")"); }
    else
    {
        clips.assertString("(next " + currentID + " " +
            choicesButtons.getSelection().getActionCommand() +
            ")");
    }
    runExpSys();
}
elseif (ae.getActionCommand().equals("Restart"))
{
    clips.reset();
    runExpSys();
}
elseif (ae.getActionCommand().equals("Prev"))
{
    clips.assertString("(prev " + currentID + ")");
    runExpSys();
}
}
```

Cette partie de code représente comment lire les réponses de client oui et non.

```
choicesPanel.removeAll();
choicesButtons = newButtonGroup();

PrimitiveValuepv = fv.getFactSlot("valid-answers");

String selected = fv.getFactSlot("response").toString();
System.out.println("--- response " +selected);
for (inti = 0; i<pv.size(); i++)
{
    PrimitiveValuebv = pv.get(i);
    System.out.println("--- " + bv.toString());
    JRadioButtonrButton;

    if (bv.toString().equals(selected))
    { rButton =
      newJRadioButton(autoResources.getString(bv.toString()), true); }
    else
    { rButton =
      newJRadioButton(autoResources.getString(bv.toString()), false); }

    rButton.setActionCommand(bv.toString());
    choicesPanel.add(rButton);
    choicesButtons.add(rButton);
}

choicesPanel.repaint();
```

Les fichiers ressources sont utilisés pour mémoriser les données questions réponses pour le système expert et en différentes langues. Ici par exemple en Anglais.

```
# Properties File for English
AutoDemo=AutoDemo
No=No
Yes=Yes
Normal=Normal
Burned=Burned
Contaminated=Contaminated
Next=Next
Prev=Prev
Restart=Restart
WelcomeMessage=WelcometotheComputerDiagnosisExpertSystem.
PowerOnQuestion=DoesthePowercomeson?
BootScrQuestion=DoestheBootScreenshowing?
PowerSrcQuestion=DoesthePowersource?
Boot2tryQuestion=DoestheBooton2ndtry?
SluggishQuestion=IstheHearbeepQuestion?
NewHardQuestion=DoesaNewHardwareinstalled?
VoltSetQuestion=Doesthe110/120voltsisset?
InstallMoboQuestion=IsthemotherBordinstalled?
UseLiveOutlet=youhavetoUseLiveOutlet.
SelectProperVoltage= youhavetoSelectProperVoltage.
VedioFailure=ItisaVedioFailure.
proceedMoBoFailure=ProceedtoMotherBordFailure
PrematurePowerOk=trydiffrentpowersupply.
```

L'implémentation des faits dans le système expert.

```
(deftemplate UI-state
  (slot id (default-dynamic (gensym*)))
  (slot display)
  (slot relation-asserted (default none))
  (slot response (default none))
  (multislot valid-answers)
  (slot state (default middle)))
(deftemplate state-list
  (slot current)
  (multislot sequence))
(deffacts startup
  (state-list))
```

Ici un exemple d'implémentation des règles dans le système expert.

```
(defrule determine-power-com-on ""
  (logical (start))
  =>
  (assert (UI-state (display PowerOnQuestion)
                  (relation-asserted power-starts)
                  (response No)
                  (valid-answers No Yes))))

(defrule determine-boot-screen ""
  (logical (power-starts Yes))
  =>
  (assert (UI-state (display BootScrQuestion)
                  (relation-asserted boot-screen)
                  (response No)
                  (valid-answers No Yes))))
```

IV.5 Conclusion

Dans ce chapitre nous avons mis en œuvre notre modèle de conception pour notre système d'évaluation des pannes informatique présenté dans le chapitre précédant à travers la génération du code Java fx et le développement de système expert par la langage CLIPS et le système chatbot par le langage AIML.

CONCLUSION ET PERSPECTIVES

Conclusion générale

Dans ce mémoire, nous nous sommes intéressés au domaine de maintenance informatique qui est un domaine très important, en raison du nombre d'ordinateurs et leur rôle dans la vie quotidienne. Notre objectif est la conception d'un outil qui facilite aux utilisateurs le diagnostic des pannes informatiques. Nous avons proposé une architecture basée sur des notions d'intelligence artificielle A savoir les systèmes experts et le chatbot.

Nous avons parlé brièvement sur le micro-ordinateur et ses composants et aussi sur le domaine de maintenance informatique, les pannes et les outils de dépannage. Par la suite Nous avons présenté l'état de l'art de notre contexte. Ensuite, on a présenté l'architecture logicielle de notre système d'évaluation des pannes informatique et sa modélisation des données et la modélisation de système par le langage UML. Par la suite, nous avons présenté la phase de réalisation qui consiste à l'implémentation de notre système ExpertBot Repair.

Notre système d'évaluation des pannes informatique est un système qui offre une communication très utile avec l'utilisateur, grâce aux points forts des systèmes experts et les agents conversationnels, Nous avons construit les bases de connaissances du système Expertbot repair par l'utilisation des arbres de décisions.

En fin, nous ne disons pas que notre travail est parfait, de ce fait, plusieurs extensions et perspectives futures pour l'architecture et l'implémentation restent à faire, les plus importants sont :

- Développer des versions orientées smartphones Android et aussi le web.
- Améliorer l'architecture de ce système pour obtenir une Plateforme destinée à divers types de diagnostics.
- Bénéficier de la nouvelle innovation de Facebook (12- 04- 2016), «The 8th Facebook Developer Conference » Facebook annoncera chatbot API pour son outil « Messenger » aux développeurs.

REFERENCES
BIBLIOGRAPHIQUES

Bibliographie

- [1] J.-N. Anderruthy, Maintenance et dépannage d'un PC en réseau, ENI, 2008 .
- [2] J.-P. FRAYSSE, Le micro-ordinateur, cybercantou, 2012.
- [3] J. Chambard, «systeme-dexploitation,» 30 08 2015. [En ligne]. Available: <http://www.dictionnaireduweb.com/systeme-dexploitation/>. [Accès le 29 02 2016].
- [4] «depannage,» infopme, [En ligne]. Available: <http://www.infopme.ch/fr/informatique/business-recovery/depannage/> . [Accès le 01 02 2016].
- [5] «bootcd,» hiren, [En ligne]. Available: <http://www.hiren.info/pages/bootcd>. [Accès le 06 03 2016].
- [6] B. Burrows, «ubcd4win,» 2004. [En ligne]. Available: www.ubcd4win.org/. [Accès le 06 04 2016].
- [7] F. Santos, «Arbres de décision,» CNRS, UMR 5199 PACEA, p. 1, 2015.
- [8] C. Mike , L. Jim et B. Eric, «Failure Diagnosis Using Decision Trees,» *IEEE Computer Society Washington, DC, USA ©2004* , pp. 36-43 , 2004 .
- [9] Narayanan, J.Mickens, M. Szummer Dushyanth, «Snitch: Interactive Decision Trees for Troubleshooting Misconfigurations,» *USENIX Association Berkeley, CA, USA* , 2007.
- [10] N. Abchiche, Systèmes intelligents, Laboratoire IBISC.
- [11] Ergado, A.Ayde, «SELF LEARNING COMPUTER TROUBLESHOOTING EXPERT SYSTEM,» *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. Vol. 7, January 2016 .
- [12] N.Dev, B.Anderson, «Pimtool, an Expert System to Troubleshoot Computer Hardware Failures.,» *the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference,,* January 1997.
- [13] Y. Bassil, «EXPERT PC TROUBLESHOOTER WITH FUZZY-LOGIC AND SELF-LEARNING SUPPORT,» *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. Vol.3, p. No.2, 2011.
- [14] W. Sylvester I. Ele and Adesola, «Design of Computer Fault Diagnosis and Troubleshooting System (CFDTS)».
- [15] Helen J. Wang, John C. Platt, Yu Chen, Ruyun Zhang, Yi-Min Wang, «Automatic Misconfiguration Troubleshooting with PeerPressure,» *OSDI'04 Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation* , vol. Volume 6, pp. Pages 17-17 , 2004.
- [16] John S. Breese, David Heckerman, and Carl Kadie, «Empirical Analysis of Predictive Algorithms for Collaborative Filtering,» *UAI'98 Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. Pages 43-52 , 1998.
- [17] H.schyns, cours maintenance informatique, France, mars 2009.

- [18] Soila P. Kavulya, K. Joshi, F. Giandomenico, P. Narasimhan, «Failure Diagnosis of Complex Systems,» *Resilience Assessment and Evaluation of Computing Systems*, pp. 239-261, 2012.
- [19] C. Stewart, T. Kelly, and A. Zhang, «Exploiting nonstationarity for performance,» *In European conference on Computer systems (EuroSys)*, p. 31–44, Lisbon, Portugal, March 2007.
- [20] R. R. Kompella, J. Yates, A. G. Greenberg, and A. C. Snoeren, «IP fault localization,» *In USENIX Symposium on Networked Systems Design and*, p. 57–70, Boston, MA, May 2005..
- [21] P. Bahl, R. Chandra, A. G. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang, «owards highly reliable enterprise network services via inference of multi-level,» *In ACM Conference on Applications, Technologies, Architectures,*, p. 13–24, Kyoto,Japan, August 2007..
- [22] Gibson, B. Schroeder and G. A, «Disk failures in the real world: What does an,» *In USENIX Conference on File and Storage Technologies*, p. 1–16, San Jose, CA, February 2007.
- [23] J. Tan, X. Pan, S. Kavulya , R. Gandhi, and P. Narasimhan, «SALSA: Analyzing Logs as State Machines.,» *In USENIX Workshop on Analysis of System Logs, SanDiego,, CA*, December 2008..
- [24] H. Kang, H. Chen, and G. Jiang, «PeerWatch: a fault detection and diagnosis tool for virtualized consolidation systems.,» *In IEEE International Conference on Automatic Computing*, p. 119–128, Washington, DC, June 2010..
- [25] Fox., E. Kiciman and A, «Detecting application-level failures in component-based internet services.,» *IEEE Transactions on Neural Networks: Special Issue on Adaptive Learning Systems in Communication Networks*, p. 1027– 1041, September 2005.
- [26] C. Yuan, N. Lao, J.-R. Wen, J. Li, Z. Zhang, Y.-M. Wang, and W.-Y. Ma, «Automated known problem diagnosis with event traces.,» *Wang, and W.-Y. Ma. Automated known problem diagnosis with event traces. In European conference on Computer systems (EuroSys)*, p. 375–388, April 2006.
- [27] chatbotmaker, «usine-laquo-chatbot-raquo,» [En ligne]. Available: <http://www.bokomarupublications.com/usine-laquo-chatbot-raquo.html>. [Accès le 16 04 2016].
- [28] J. Weizenbaum, «ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine».
- [29] V. Cerf, « PARRY Encounters the DOCTOR,» *Network Working Group*, 21 January 1973.
- [30] T. Winograd, *Understanding Natural Language*, 1972 .
- [31] [En ligne]. Available: <http://www.alicebot.org/about.html>. [Accès le 12 03 2016].
- [32] «Messenger Platform,» Facebook , [En ligne]. Available: <https://developers.facebook.com/docs/messenger-platform>. [Accès le 16 04 2016].
- [33] CLIPS, «A Tool for Building Expert Systems,» [En ligne]. Available: <http://clipsrules.sourceforge.net/>. [Accès le 20 04 2016].
- [34] M. Rosenthal, *Boot failure troubleshooting flowchart*, 2003.

- [35] S. expert. [En ligne]. Available: <http://www.tripalium.com/fiches/evaluation/Sysexpert.html>. [Accès le 10 03 2016].
- [36] E, Dr. Richard S. WALLAC, The Elements of AIML Style, ALICE A. I. Foundation, Inc., March 28, 2003.
- [37] «JavaFX,» [En ligne]. Available: docs.oracle.com/javafx/. [Accès le 5 4 2016].