

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

Université d'Ibn Khaldoun – Tiaret  
Faculté des Mathématiques et de l'Informatique  
Département Informatique



## Mémoire de fin d'études

Thème :

# *Modélisation de la coopération des intervenants dans la e-maintenance.*

Pour l'obtention du diplôme de Master II  
Spécialité : Génie Informatique  
Option : Système d'Information

Réalisé par : DJILALI Abdelkirm  
MERDJET YAHIA Bengamra

Dirigé par : Si abdelhadi Hmida

Année Universitaire 2015 / 2016

# *Remerciement*

En préambule de ce mémoire, nous souhaitons adresser tous nos remerciement aux personnes qui nous ont apporté leurs aide et qui ont ainsi contribué à l'élaboration de ce mémoire.

Nous tenant à remercier Si ABDELHADI Hmida, notre tuteur, pour son aide lors de l'élaboration de ce travail et son suivi durant la finalisation de ce projet.

Nous adressons nos plus sincères remerciements à nos parents, nos proches et amis qui nous ont toujours soutenu et encouragés et Hafidh au cours de la réalisation de ce mémoire.

# Dédicace

En reconnaissance de tous les sacrifices consentis, encouragement et patience par chacun pour me permettre d'atteindre cette étape de ma vie, ce mémoire est dédié exceptionnellement à mes chers parents.

Je dédie ce mémoire

À ma grande mère, mes frères et sœurs et à toute ma famille.

À mes collègues et mes amis une mention spéciale, Oussama, Habib, Hafidh et Yacine et Amine

Abdelkrim Djilali

# Dédicace

C'est avec profonde gratitude et sincères mots, que je dédie ce modeste travail à mes chers parents, qui ont sacrifié leur vie pour notre réussite et j'ai éclairé mon chemin par leurs conseils judicieux.

J'espère qu'un jour, je pourrai leurs rendre un peu de ce qu'ils ont fait pour moi,

Que dieu leur procure bonne santé et longue vie.

**A la lumière de mes jours, la source de mes efforts, la flamme de Mon cœur,  
ma vie et mon bonheur ; MAMAN que j'adore.**

Je dédie aussi ce travail à mes frères et sœurs, toute ma famille.

À mes amis : Amine A, Hafidh K, Yassine R, Oussama S, Younes, Chenwi, Salim S,

Abdelhak, Hisham, Redha et Fouzi H

À tous ceux qui sont si chers pour moi.

A mon binôme Djilali Abdelkrim, à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

**MERDJET YAHIA** Bengamera

## **Summary**

The work presented in this thesis consists in the establishment of a negotiation and communication multi-agent System. The application relates to a maintenance system, therefore, we define a model describing the behavior of three types of agents: Equipment Agent, Agent Resource (repairman) and Operator agent latter is considered the link between the user and the system it comes to configuring cited above, and it is considered as an administrator who manages all system properties. As for resource agents they work as contractors for repair tender launched by the Equipment agent.

### **Key words**

Maintenance, E-maintenance, S-Maintenance, SMA, Multi Agent System, Negotiation, Communication, criteria, deadlines, costs, Gantt, Jade.

## **Résumé**

Le Travail présenté dans ce mémoire consiste en la mise en place d'un system de négociation et communication multi agent. L'application porte sur un système de maintenance, de ce fait, nous définissons un modèle en décrivant le comportement de trois types d'agents : Agent Equipement, Agent Ressource (réparateur) et l'agent Operateur ce dernier est considéré comme le lien entre l'utilisateur et le système, il intervient pour la configuration des cité en dessus, ainsi il est considéré en tant qu'un administrateur qui gère l'ensemble des propriétés du système. Quant aux agents Ressource ils interviennent comme des contractants pour un appel d'offre de réparation lancé par l'agent Equipement.

### **Mots clés**

Maintenance, E-maintenance, S-Maintenance, Agent, SMA, Système multi agent, Négociation, Communication, critères, délais, couts, Gantt, Jade.

## Table de Matières :

|                             |    |
|-----------------------------|----|
| Remerciement .....          | 2  |
| Dédicace .....              | 3  |
| Résumé .....                | 5  |
| Mots clés .....             | 5  |
| Introduction générale ..... | 11 |

### Chapitre I : La Maintenance

|  |    |
|--|----|
| Introduction .....   | 13 |
| 1. La Maintenance .....  | 13 |
| 1.1 Définition .....   | 13 |
| 1.2 Types de maintenance .....                                   | 13 |
| 1.3 La Maintenance corrective .....                              | 14 |
| 2 Les systèmes de maintenance .....                              | 15 |
| 2.1 GMAO - Gestion de Maintenance Assistant par Ordinateur ..... | 15 |
| 2.1.1 Définition .....   | 16 |
| 2.1.2 Fonctionnalités d'une GMAO .....                           | 16 |
| 2.1.3 Utilisation .....  | 16 |
| 2.1.4 Les systèmes GMAO existants : (CMMS, s.d.) .....           | 17 |
| 2.1.5 L'Architecture de GMAO .....                               | 17 |
| 2.2 La Télémaintenance .....                                     | 18 |
| 3.2.1 Définition .....   | 18 |
| 3.2.2 L'Architecture de la télémaintenance .....                 | 18 |
| 3.2.3 Exemple de système de télémaintenance .....                | 19 |
| 3.3 E-MAINTENANCE .....  | 20 |
| 3.3.1 Présentation .....   | 20 |
| 3.3.3 Définition .....   | 20 |
| 3.3.4 Domaines d'application de e-maintenance .....              | 20 |
| 3.3.5 Plate-forme e-maintenance .....                            | 20 |
| 3.3.6 Architecture .....   | 20 |
| 3.3.7 Plateformes d'e-maintenance existantes .....               | 21 |
| 3.3.8 Plateforme de projet .....                                 | 21 |
| 3.3.9 Plateforme académique .....                                | 25 |
| 3.3.10 Les limites de l'e-maintenance .....                      | 27 |
| 3.4 S-Maintenance .....  | 27 |

|  |    |
|--|----|
| 3.4.1 Définition .....                                   | 27 |
| 3.4.2 Définition de la plateforme de s-maintenance ..... | 28 |
| 3.4.3 Architecture de la s-maintenance .....             | 28 |
| 3.4.4 Système Existant .....                             | 28 |
| Fonctionnement Du Système .....                          | 31 |
| Fonctionnalité .....                                     | 31 |
| Conclusion .....   | 33 |

## Chapitre II : Les Systèmes Multi-Agents

|  |    |
|--|----|
| Introduction .....                                       | 35 |
| 1) Les Agents .....                                      | 35 |
| 1.1 Définitions .....                                    | 35 |
| 1.2 - Caractéristiques d'un agent .....                  | 36 |
| 1.3 - Catégories d'agents .....                          | 37 |
| 1.4 - Architectures d'agents .....                       | 37 |
| 1.4.1 Agent basé sur la logique .....                    | 37 |
| 1.4.2 Agent délibératif .....                            | 37 |
| 1.4.3 Agent BDI (Belief-Desire-Intention) .....          | 37 |
| 1.4.4 Agents Cognitives et réactives .....               | 38 |
| 1.4.5 Architecture hybride ou architecture couches ..... | 39 |
| 2) Les Systèmes Multi-agents .....                       | 40 |
| 2.1 Définitions .....                                    | 40 |
| 2.2 Caractéristiques des systèmes multi-agents .....     | 40 |
| 2.3 La communication entre les agents .....              | 41 |
| 2.4 Coopération .....                                    | 41 |
| 2.6 Plateformes multi agents .....                       | 42 |
| 2.6.1 Critères de sélection .....                        | 43 |
| 2.6.2 L'évaluation .....                                 | 45 |
| 2.6.3 La plate-forme JADE .....                          | 46 |
| 2.7 Exemples d'applications .....                        | 47 |
| 2.8 Modèle organisationnel .....                         | 48 |
| 2.8.1 ALAADIN : Agent-Groupe-Rôle .....                  | 48 |
| Conclusion .....   | 50 |

## Chapitre III : Modélisation et Conception

|                             |    |
|-----------------------------|----|
| Introduction .....          | 52 |
| 1. Le modèle Aalaadin ..... | 53 |

|       |  |    |
|-------|--|----|
| 1.1   | Présentation .....   | 53 |
| 1.2   | Atouts du modèle Aaladin .....   | 53 |
| 1.3   | Apports et limitations du modèle Aalaadin .....                          | 54 |
| 2     | Le modèle agent groupe rôle .....  | 55 |
| 2.1   | Agent .....  | 55 |
| 2.2   | Le Groupe .....  | 56 |
| 2.3   | Le Rôle .....  | 57 |
| 2.3.1 | Rôle de l'agent d'Equipement .....                                       | 58 |
| 2.3.2 | Rôle de l'agent ressource .....  | 58 |
| 2.3.3 | Rôle de l'agent Opérateur .....  | 58 |
| 3.    | Le modèle d'interaction.....   | 59 |
| 4.    | Proposition d'un modèle de pilotage basé sur l'interactivité .....       | 59 |
| 4.1   | Les fonctions du pilotage .....  | 60 |
| 4.2   | Architecture de pilotage .....   | 61 |
| 5.    | Protocole d'interaction.....   | 62 |
| 5.1   | Le protocole de communication .....                                      | 62 |
| 5.2   | Protocole de négociation .....   | 63 |
| 6.    | Le lien entre le rôle et l'interaction dans le domaine du pilotage ..... | 65 |
| a)    | - Les rôles élémentaires (niveau 1) .....                                | 65 |
| b)    | - Les rôles complexes (niveau 2) .....                                   | 66 |
| c)    | - Les rôles émergents (niveau 3) .....                                   | 66 |
|       | Conclusion .....   | 67 |

## Chapitre IV : Réalisation & Implémentation

|     |  |    |
|-----|--|----|
|     | Introduction .....                                       | 69 |
| 1.  | Les outils utilisés.....                                 | 69 |
| 1.1 | Langage utilisé .....                                    | 69 |
| 1.2 | IDE Utilisé .....  | 69 |
| 1.3 | Serveur Utilisé .....                                    | 70 |
| 1.4 | Connecteur JDBC (Java Database Connectivity) MySQL ..... | 70 |
| 1.5 | SGBD .....   | 70 |
| 1.6 | Agent Sniffer .....                                      | 71 |
| 1.7 | Les bibliothèques Utilisé .....                          | 71 |
| a.  | JDBC .....   | 72 |
| b.  | JADE .....   | 72 |
| c.  | JFreeChart .....   | 72 |

|   |           |
|---|-----------|
| <b>2. Etapes de Conception du notre système</b> ..... | <b>73</b> |
| <b>1. Création des bases de données</b> .....         | <b>73</b> |
| <b>2. Présentation</b> .....                          | <b>75</b> |
| <b>La partie (A)</b> .....                            | <b>75</b> |
| <b>3. Fonctionnalité</b> .....                        | <b>76</b> |
| <b>La partie (B)</b> .....                            | <b>76</b> |
| <b>La partie (C)</b> .....                            | <b>80</b> |
| <b>La partie (D)</b> .....                            | <b>81</b> |
| <b>4. Modèle d'utilisation du système conçu</b> ..... | <b>82</b> |
| <b>Conclusion</b> .....                               | <b>83</b> |
| <b>Conclusion Général</b>                             |           |
| <b>Références</b> .....                               | <b>85</b> |

## Table des Figures :

### Chapitre I : La Maintenance

|  |           |
|--|-----------|
| <b>FIGURE 1:</b> TYPE DE MAINTENANCE (AFNOR) .....                             | <b>14</b> |
| <b>FIGURE 2 :</b> SYSTEME DE MAINTENANCE.....                                  | <b>17</b> |
| <b>FIGURE 3:</b> ARCHITECTURE DE TELEMANTENANCE .....                          | <b>18</b> |
| <b>FIGURE 4:</b> SYSTEME DE TELEMANTENANCE SELECTRON .....                     | <b>19</b> |
| <b>FIGURE 5:</b> ARCHITECTURE DU CONCEPT DE E-MAINTENANCE .....                | <b>21</b> |
| <b>FIGURE 6:</b> ARCHITECTURE DE LA PLATEFORME MIMOSA.....                     | <b>22</b> |
| <b>FIGURE 7:</b> ARCHITECTURE DE LA PLATEFORME PROTEUS .....                   | <b>23</b> |
| <b>FIGURE 8:</b> ARCHITECTURE DE LA PLATEFORME PROMISE.....                    | <b>24</b> |
| <b>FIGURE 9:</b> ARCHITECTURE DE LA PLATEFORME SMMART .....                    | <b>25</b> |
| <b>FIGURE 10:</b> ARCHITECTURE DE LA PLATEFORME ORIENTE AGENT .....            | <b>26</b> |
| <b>FIGURE 11:</b> ARCHITECTURE DU CONCEPT DE S-MAINTENANCE .....               | <b>28</b> |
| <b>FIGURE 12:</b> ARCHITECTURE DE LA PLATEFORME DE S-MAINTENANCE PROPOSE ..... | <b>32</b> |

### Chapitre II : Les Systèmes Multi-Agents

|  |           |
|--|-----------|
| <b>FIGURE 1:</b> FONCTIONNEMENT D'UN AGENT .....               | <b>35</b> |
| <b>FIGURE 2:</b> PROPRIETES D'UN AGENT .....                   | <b>36</b> |
| <b>FIGURE 3 :</b> L'ARCHITECTURE D'UN AGENT BDI.....           | <b>38</b> |
| <b>FIGURE 4 :</b> LES AGENTS COGNITIFS .....                   | <b>39</b> |
| <b>FIGURE 5 :</b> LES AGENTS REACTIFS .....                    | <b>39</b> |
| <b>FIGURE 6 :</b> L'ARCHITECTURE D'UN AGENT MULTI-NIVEAUX..... | <b>39</b> |
| <b>FIGURE 8:</b> LES TROIS CONCEPTS CENTRAUX .....             | <b>48</b> |

### Chapitre III : Modélisation et Conception

|  |    |
|--|----|
| <b>FIGURE 1:</b> NOTION AGENT GROUPE ROLE AALADIN .....  | 55 |
| <b>FIGURE 2 :</b> NOTION DE GROUPE AALADIN .....   | 57 |
| <b>FIGURE 3 :</b> ROLE AGENT OPERATEUR .....   | 58 |
| <b>FIGURE 4:</b> FONCTION PILOTAGE DU SYSTEME DE MAINTENANCE .....   | 60 |
| <b>FIGURE 5 :</b> PILOTAGE DU SYSTEME DE MAINTENANCE AVEC MODULES DE COMMUNICATIONS ET DE NEGOCIATIONS ..... | 61 |
| <b>FIGURE 6 :</b> COMMUNICATION SOUS PROTOCOLE FIPA REQUEST .....  | 63 |
| <b>FIGURE 7 :</b> NEGOCIATION SOUS PROTOCOLE CNP .....   | 64 |

## Chapitre IV : Réalisation & Implémentation

|   |    |
|---|----|
| <b>FIGURE 1:</b> JDBC CONNECTER .....                               | 70 |
| <b>FIGURE 2:</b> AGENT SNIFFER .....                                | 71 |
| <b>FIGURE 3:</b> JDBC CONNECTEUR .....                              | 72 |
| <b>FIGURE 4:</b> JADE LIB .....                                     | 72 |
| <b>FIGURE 5:</b> JFREECHART LIB .....                               | 72 |
| <b>FIGURE 6:</b> BASE DE DONNEES " MAINTENANCE" .....               | 73 |
| <b>FIGURE 7:</b> INTERFACE PRINCIPALE .....                         | 75 |
| <b>FIGURE 8:</b> BARRE DE MENU .....                                | 75 |
| <b>FIGURE 9:</b> COMMUNICATION & NEGOCIATION .....                  | 76 |
| <b>FIGURE 10:</b> LES ONGLETTES DE L'APPLICATION .....              | 76 |
| <b>FIGURE 11:</b> EQUIPEMENT .....                                  | 77 |
| <b>FIGURE 12:</b> L'AJOUTE D'UN NOUVEL EQUIPEMENT A MAINTENIR ..... | 77 |
| <b>FIGURE 13:</b> AJOUTER NOUVELLE RESSOURCE .....                  | 78 |
| <b>FIGURE 14:</b> ONGLET DES RESSOURCES .....                       | 78 |
| <b>FIGURE 15:</b> GESTION DES PANNES .....                          | 78 |
| <b>FIGURE 16:</b> MODIFIER UNE PANNE .....                          | 79 |
| <b>FIGURE 17:</b> AJOUTER UNE PANNE .....                           | 79 |
| <b>FIGURE 18:</b> GESTION DES PIECES .....                          | 79 |
| <b>FIGURE 19:</b> AJOUTER UNE PIECE .....                           | 80 |
| <b>FIGURE 20:</b> MODIFIER UNE PIECES .....                         | 80 |
| <b>FIGURE 21:</b> HISTORIQUE .....                                  | 80 |
| <b>FIGURE 22:</b> REPARATIONS ACTUELS .....                         | 81 |
| <b>FIGURE 23:</b> DIAGRAMME DE GANTT .....                          | 81 |
| <b>FIGURE 24 :</b> MODELE D'UTILISATION DU PROTOTYPE .....          | 82 |

## Liste des Tableaux :

|   |    |
|---|----|
| <b>TABLEAU 1:</b> LES DIFFERENTS TYPES D'AGENT.....                         | 38 |
| <b>TABLEAU 2:</b> LES RESULTATS DE L'EVALUATION .....                       | 45 |
| <b>TABLEAU 1:</b> LES PERFORMATIFS D'ACL UTILISES POUR LA NEGOCIATION ..... | 65 |

## Introduction Générale

Aujourd'hui, trois forces agissent sur les entreprises, Les clients qui prennent le pouvoir, La concurrence qui se durcit et Le changement qui devient incessant Ces trois forces réclament flexibilité, réactivité. Dans le contexte industriel actuel qui se démarque fortement des processus de maintenance de masse (Taylorisme), l'entreprise doit être considérée, non plus comme une organisation centralisé, où peu de place est laissée à l'autonomie et peu d'importance est accordée aux interactions entre différents composants, mais comme un ensemble de systèmes de prise de décision, de traitements et de communications. Cela revient à dire une organisation décentralisée, dynamique et réactive. Ce qui se traduit par une décentralisation de la prise de décision aux différents niveaux de l'organisation. Les structures centralisées sont parfaitement capables de mener à bien des tâches qui sont déterminées à l'avance, mais lorsqu'il faut traiter avec souplesse de nouveaux problèmes qui surgissent à un rythme rapide, elles ne sont pas en mesure de le faire. Sur la base de ce paradigme, les concepts informatiques vont permettre de modéliser cette organisation « dynamique » et « décentralisée ». Par ce travail on a vas essayer de montrer que l'approche multi agents permet de définir un nouveau type de modèle. Par le niveau d'abstraction et ses fondements théoriques, cette approche est bien adaptée pour modéliser la dynamique de l'organisation des systèmes de maintenance, et à l'étude des changements organisationnels qui s'y produisent. Notre travail est accès sur les systèmes de pilotage des systèmes de maintenance. Plus précisément des systèmes de pilotage basés sur la technologie agent. La conception et la réalisation de ces systèmes ne sont pas une tâche aisée La réalisation de ce projet consiste à comprendre les particularités des systèmes de maintenance et les fonctions de pilotage des systèmes industriels plus les concepts fondamentaux de la programmation multi agent. A travers ce projet nous allons essayer de mettre en place un système de maintenance qui possède un certain degré d'autonomie au niveau des interactions.

# Chapitre I

---

La Maintenance

### Introduction :

L'évolution et la complexité des systèmes de production ont induit la volonté des entreprises à augmenter la rentabilité et d'optimiser les couts et les frais liés à la production , ce qui oblige les industriels à structurer et organiser le plus important support de la production à savoir : la maintenance .

Afin que ce support soit efficace sur le plan productif et couts, de nouveaux concepts et de nouvelles procédures d'intervention basées sur les moyens technologiques doivent être mise en place.

Dans le présent chapitre nous allons décrire les types et systèmes de maintenance existants en illustrant la structure de chaque système.

## 1. La Maintenance:

### 1.1 Définition:

La définition normative de la maintenance est la suivante :

« ensemble des actions permettant de maintenir ou de rétablir un bien dans un état spécifié ou en mesure d'assurer un service déterminé »

(Selon l'Association Française de Normalisation AFNOR Il existe 4 types de maintenance (AFNOR, s.d.)

### 1.2 Types de maintenance:

- **Préventive** : anticipation de panne, prévision ;  
Ce type de maintenance est concentré sur la prévision des pannes dans l'intention de réduire la probabilité de défaillance d'un bien, d'un équipement ou la dégradation d'un service rendu. On change les pièces préventivement, conformément à leur durée de vie. La maintenance préventive distingue trois formes :
  - **la maintenance systématique** : c'est une maintenance préventive effectuée suivant un échancier établi selon le temps ou le nombre d'unités
  - **la maintenance conditionnelle** : pour les équipements importants, c'est une maintenance préventive ou les opérations sont déclenchées que lorsque certains paramètres mesurables atteignent un seuil limite

- **la maintenance prévisionnelle** : une maintenance préventive ressemble à la maintenance conditionnelle mais exécuter en suivant extrapolées de l'analyse et de l'évaluation des paramètres significatifs de la dégradation du bien.
- **Proactive** : La maintenance pro-active ressemble à la maintenance préventive mais en tenant compte de l'historique des pannes. Cette maintenance repose sur l'exploitation du retour d'expérience et sur l'analyse approfondie des phénomènes à l'origine des défaillances.
- **Améliorative** : fait partie de la grande maintenance, elle introduit des améliorations dans les fonctionnalités et/ou de la fiabilité d'un équipement.
- **Corrective** : dépannage, réparation après défaillance. Il s'agit ici de remettre en état un équipement qui a cessé de fonctionner de manière inattendue.

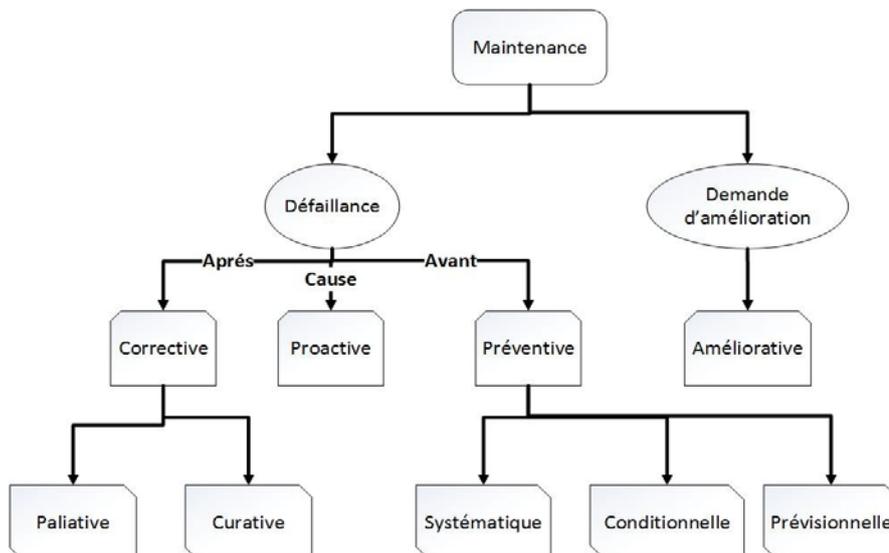


Figure 1 : Type de maintenance (AFNOR)

### 1.3 La Maintenance corrective:

Ce qui est visée ici c'est la maintenance corrective, elle consiste à intervenir sur un équipement une fois que celui-ci est défaillant les dégradations sont corrigées afin de maintenir ou de rétablir sa conformité aux spécifications ; Ainsi lorsqu'une panne est détectée, l'équipement sera remis dans un état lui permettant de remplir les fonctions pour lesquelles il a été conçu.

La maintenance corrective est surtout utilisée lorsque l'indisponibilité du matériel n'a pas de conséquences majeures sur le processus de production ou quand les contraintes de sécurité sont faibles ; Car elle nécessite l'immobilisation temporaire du matériel ; l'intérêt est qu'elle ne ralentit pas et ne bloque pas ce processus, aussi que les conséquences sur la sécurité doivent être faibles [(AFNOR, s.d.)].

Elle comprend deux types de dépannage :

### **Maintenance palliative** -type réparation provisoire- :

Consiste à intervenir pour dépanner un équipement pour que ce dernier puisse reprendre une partie de ces fonctionnalités et devient opérationnel, cette intervention est provisoire et nécessite un suivi par une action.

Consiste à dépanner un équipement qui le nécessite, afin que tout ou partie de ses fonctions soit opérationnel. Elle est provisoire et est nécessairement suivie d'une action curative dans les plus brefs délais.

### **Maintenance curative** -type de réparation permanent- :

Son objet est de réparer l'équipement et de le remettre dans son état d'origine ; Elle corrige ses défauts afin qu'il fonctionne correctement, elle n'est pas provisoire. Contrairement à la maintenance palliative, elle instaure une certaine durabilité au niveau de la réparation.

## 2 Les systèmes de maintenance:

Les systèmes de maintenance sont des systèmes informatiques qui ont pour but d'aider l'entreprise à gérer les processus de maintenance de la manière la plus efficace possible (Rasovska, 2006). Les stratégies adoptées par ces systèmes intègrent tout ou partie des types de maintenance.

### 2.1 GMAO - Gestion de Maintenance Assistant par Ordinateur :

Plusieurs méthodes de la maintenance ont été informatisées et automatisées grâce à l'informatisation de la gestion de l'entreprise. Les équipements, les interventions de maintenance, la gestion des stocks, les plans et les schémas ont été informatisés par la création des fichiers informatiques. Le progiciel de gestion de maintenance assisté par ordinateur (GMAO) a permis l'intégration de ces fichiers, et l'automatisation de ces activités ainsi que les interventions quotidiennes de maintenance.

L'informatisation et l'automatisation de la gestion des entreprises a permis d'informatiser plusieurs procédures de maintenance.

Les fichiers informatiques, des équipements, des interventions, des stocks, des plans et schémas etc... ont ainsi été créés. L'intégration de ces fichiers et l'automatisation des activités de la maintenance ont été possibles grâce aux progiciels de GMAO (Gestion de Maintenance Assistée par Ordinateur).

Les événements quotidiens de la maintenance ont été traités : la panne, l'exécution du préventif, la gestion des stocks. (Rasovska, 2006)

## Chapitre I: La Maintenance

---

### 2.1.1 Définition

La **gestion de la maintenance assistée par ordinateur** (souvent abrégée en **GMAO**) est une méthode de gestion assistée d'un logiciel destiné aux services de maintenance d'une entreprise afin de l'aider dans ses activités (GMOA, s.d.)

GMAO est l'acronyme de Gestion de Maintenance Assistée par Ordinateur. C'est l'outil qui permet de gérer la traçabilité des actions de maintenance.

La GMAO est renseignée par les techniciens de maintenance à chaque fois qu'ils effectuent une opération de maintenance. (Herrman, 2010)

### 2.1.2 Fonctionnalités d'une GMAO

Une GMAO vise en premier lieu à assister les services maintenance des entreprises dans leurs missions. Un service de maintenance, selon la définition de l'AFNOR, cherche à maintenir ou à rétablir un bien (équipement) dans un état spécifié afin que celui-ci soit en mesure d'assurer un service déterminé. Une GMAO peut également être utile dans d'autres services de l'entreprise, comme la production ou l'exploitation (afin de fournir des informations sur l'état des équipements), de cette façon l'organe de décision de l'entreprise en ayant ses indicateurs ce qui lui facilite les prises de décisions en matière de renouvellement du parc matériel (GMOA, s.d.)

### 2.1.3 Utilisation

La GMAO est utilisée dans :

- Planification des tâches
- l'affectation de personnel
- réservation des matériaux
- l'enregistrement des couts
- le suivi des informations pertinentes telles que la cause du problème (le cas échéant), les temps d'arrêt en cause (le cas échéant)
- recommandations pour l'action future.

Typiquement, la GMAO fait partie de la maintenance préventive programmée automatiquement sur la base des plans d'entretien et/ou des relevés de compteur.

Plusieurs logiciels utilisent des techniques différentes pour signaler quand un travail doit être effectué.

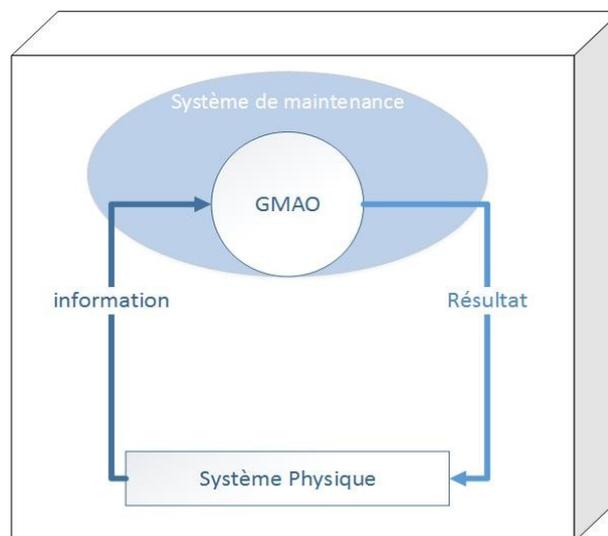
### 2.1.4 Les systèmes GMAO existants :

(CMMS, s.d.)

|                        |               |
|------------------------|---------------|
| IFS                    | DirectLine    |
| MicroMain              | Proteus       |
| Corrigo                | Landport      |
| NorthWrite             | FaciliWorks   |
| Maintenance Connection | eWorkOrders   |
| DPSI                   | NetFacilities |
| MPulse                 | Que Center    |
| Maxpanda               | Web Work      |
| eMaint                 | FastMaint     |
| Bigfoot                | ERPortal      |

### 2.1.5 L'Architecture de GMAO:

Une GMAO (Gestion de Maintenance Assistant par Ordinateur) Comprend un seul système informatique présent sur le site de production et utilisé sur le site de maintenance. Ce système est autonome sans échange de données avec d'autres systèmes. En parallèle avec la classification des entreprises, cela correspond à l'entreprise traditionnelle, donc nous parlons d'une architecture traditionnelle d'un système d'information.



*Figure 2 : Système de maintenance*

### 2.2 La Télémaintenance :

#### Définition

C'est la maintenance d'une unité fonctionnelle, assurée par la télécommunication directe entre cette unité et un centre spécialisé, c'est un service à distance d'aide au diagnostic et à la réparation.

Un Système de télémaintenance est constitué d'au moins deux systèmes informatiques un émetteur et un récepteur de données et d'informations qui échangent à distance.

Nous parlons d'une architecture distribuée, basée sur la notion de distance qui permet de transférer les données par une radio, une ligne téléphonique ou par l'intermédiaire d'un réseau local.

#### 3.2.1 L'Architecture de la télémaintenance :

Un système de télémaintenance est constitué de deux ou plusieurs systèmes ou sous- systèmes éloignés l'un de l'autre qui communiquent et échangent des données entre eux. L'un des systèmes fonctionne comme un système d'acquisition de données, il représente l'émetteur automatique de données. Le deuxième système est le récepteur, fonctionnant comme un système de traitement de données. Les données doivent être structurées de façon qu'elles puissent être acceptables par les deux systèmes.

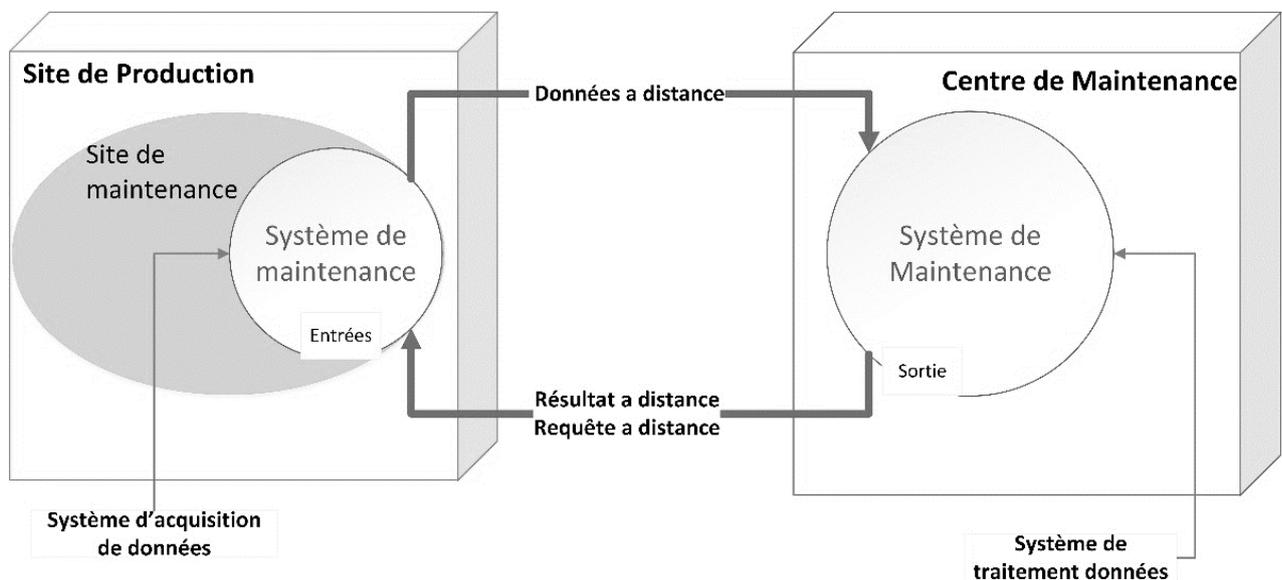
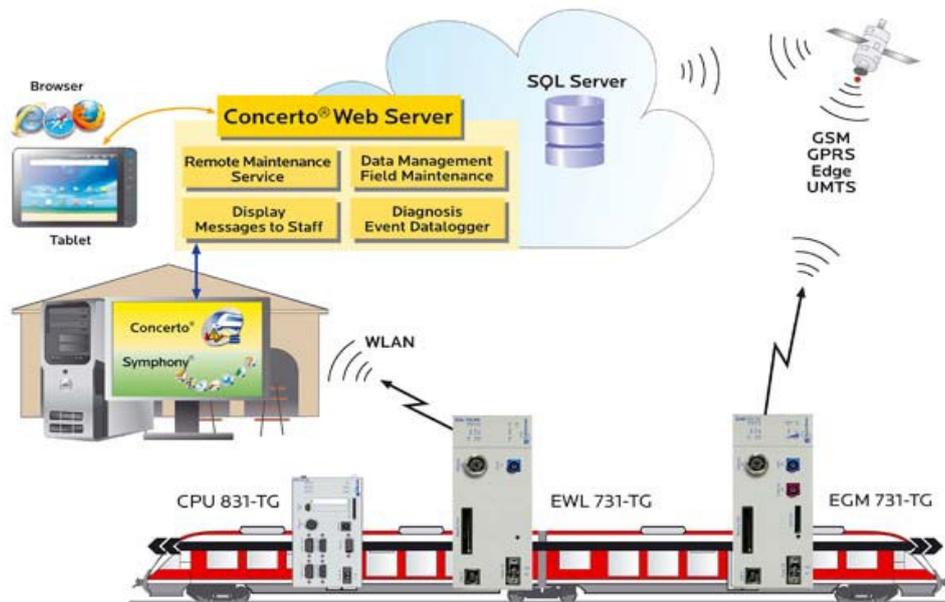


Figure 3: Architecture de Télémaintenance

### 3.2.2 Exemple de système de télémaintenance :

La société **SELECTRON** (une société suisse spécialisée dans la construction des trains, metro...). Propose un système de télémaintenance consigner des données de diagnostics automatiquement ou manuellement en temps réel en tout confort à partir des bureaux, (<http://www.selectron.ch/en/solutions/remote-maintenance.php>, s.d.)



*Figure 4: Système de télémaintenance SELECTRON*

#### Les modules de fonctionnement :

- Envoi manuel ou automatique de données définissables
- Envoi automatique de courriers électroniques à partir de la commande
- Envoi automatique de textos (SMS) à partir de la commande par exemple en cas d'alerte

### 3.3 E-MAINTENANCE

#### 3.3.1 Présentation

Emergé au début de l'année 2000, le terme maintenance est très présent dans la littérature liée à la maintenance. La e-maintenance est une forme plus évaluée de la maintenance qui permet aux différents acteurs et intervenant de partager la connaissance et de collaborer.

#### 3.3.3 Définition

E-Maintenance est un domaine multidisciplinaire basée sur l'entretien et les technologies de l'information et de la communication (TIC) veiller à ce que les services d'e-maintenance sont alignés avec les besoins et les objectifs d'affaires des clients et des fournisseurs pendant tout le cycle de vie du produit (MIRA KAJKO-MATTSSON<sup>1</sup>, 2011) [Essential Component].

#### 3.3.4 Domaines d'application de e-maintenance

- L'aviation
- Papeterie
- Automobile
- L'industrie et la construction
- Transport maritime
- L'énergie nucléaire
- énergie éolienne
- réseau ferroviaire

#### 3.3.5 Plate-forme e-maintenance

Une plateforme de coopération est définie comme un environnement de travail composé de différents composants logiciels et matériels ou ces différents composants agissent dans un cadre précis pour atteindre ces objectifs.

D'après la définition ci-dessus et la définition de la e-maintenance une plateforme de e-maintenance peut être définie étant une plateforme coopérative offrant des services permettant aux acteurs, de la e-maintenance une communication et une collaboration de travail pour la réalisation du processus de maintenance.

#### 3.3.6 Architecture

Le système d'e-maintenance est implémenté sur une plateforme distribuée coopérative intégrant différents systèmes et applications de maintenance. Cette plateforme prend appui sur le réseau d'Internet et la technologie web permet l'échange, le partager et la distribution des données et des informations.

Le principe consiste à intégrer l'ensemble des différents systèmes de maintenance dans un seul système d'information L'interopérabilité<sup>1</sup> est indispensable pour que les différents

---

<sup>1</sup> L'interopérabilité : est la capacité que possède un système informatique à fonctionner avec d'autres produits ou systèmes informatiques

systèmes dans la plate-forme puissent communiquer et s'échanger d'information de façon non ambiguë. (Ivana Rasovska, 2007)

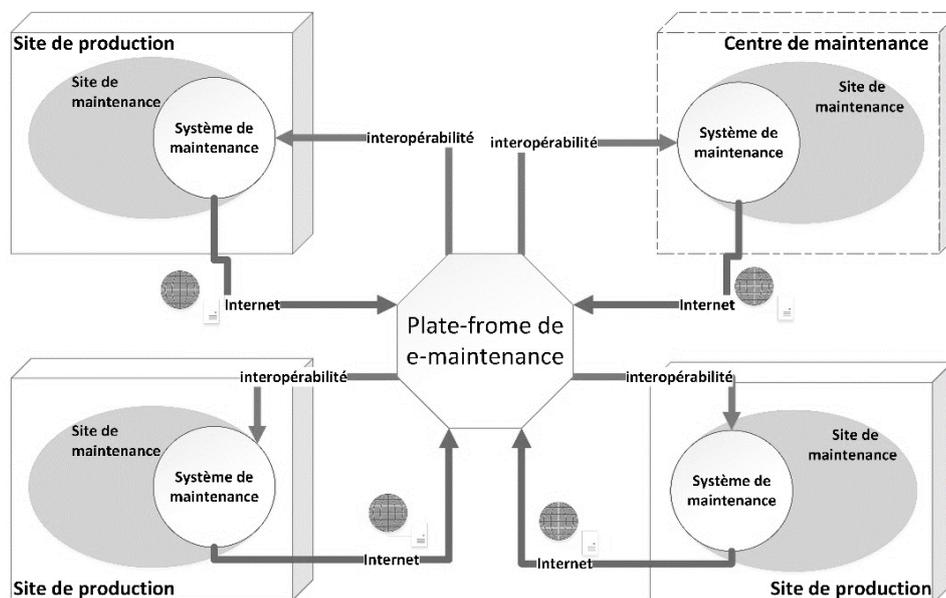


Figure 5: Architecture du concept de e-maintenance

### 3.3.7 Plateformes d'e-maintenance existantes

De nos jours plusieurs plateformes de e-maintenance sont opérationnelles classées en quatre types. Nous allons étudier dans cette section deux types de plateformes à savoir des plateformes de projets instanciés entre des industriels et des universitaires et des plateformes académiques lancées par des universitaires et des groupes de chercheurs. (Karray, 2012)

### 3.3.8 Plateforme de projet

Les plateformes citées ci-dessous sont des plateformes qui ont été conçues et réalisées dans le cadre d'un projet tels MIMOSA (projet international) et (projet européen) ESPRIT-REMAFEX, PROTEUS, PROMISE, SAMMART, et DYNAMITE.

#### **MIMOSA**

MIMOSA (The Machinery Information Management Open Systems Alliance) l'initiative était de créer un système d'information unique pour la gestion du processus de la maintenance dont l'objectif est de mise en place d'un réseau de collaboration de maintenance sous la norme open[] du protocole EAI<sup>2</sup> (Enterprise Application Integration). Des caractéristiques d'intégration d'information ont été développées pour permettre le contrôle de la valeur ajoutée intégrées et orientées vers l'industrie. Une architecture fonctionnelle OSA/CBM<sup>3</sup> (Open System Architecture for Condition-Based Maintenance) dédiée au

<sup>2</sup>EAI : est un terme informatique d'entreprise pour les plans, les méthodes et les outils visant à moderniser, consolider et coordonner l'ordinateur

<sup>3</sup>OSA/CBM : est une norme de développement pour le système de surveillance ouverte, intégrée, basée sur l'état. Il spécifie un ensemble d'interface de programmation d'application

développement de stratégies de maintenance conditionnelle ou prévisionnelle à été développée dans le cadre de

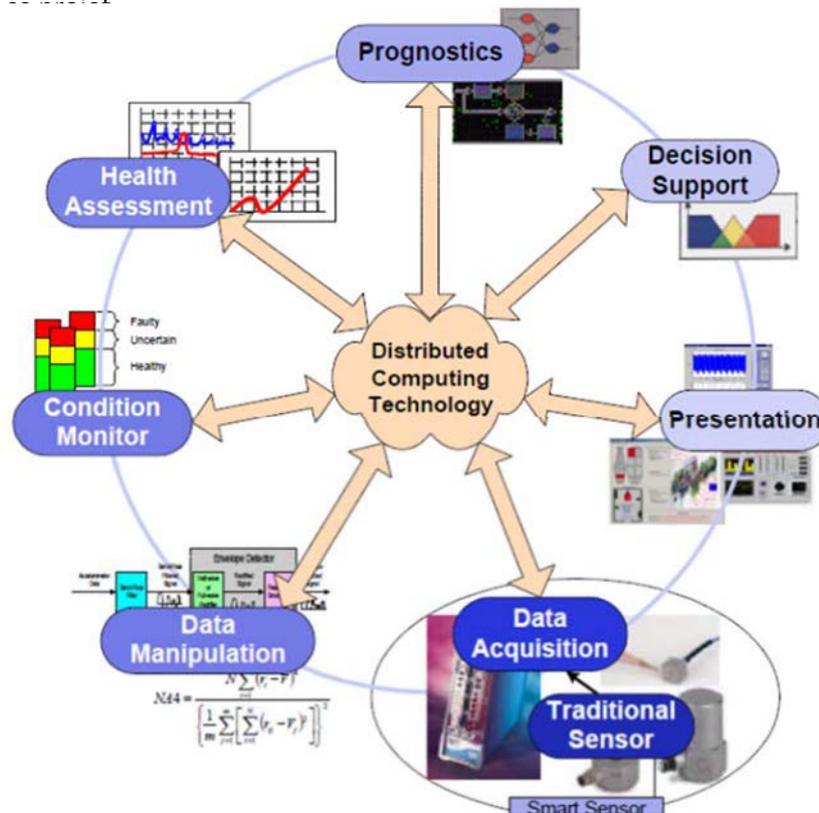


Figure 6: Architecture de la plateforme MIMOSA

### POMAESS

Proposée dans le cadre du projet ESPRIT-REMAFEX<sup>4</sup> est une plate-forme de e-maintenance basée sur les systèmes multi-agents fournissant un aide à la décision POMAESS c'est l'acronyme de « Problem-Oriented Multi-Agent based E-Service System). Chaque agent du système est un agent expert de résolution de problèmes dans différents localisation et possédant de différentes connaissances.

L'objectif principales est de créer un système qui permet aux agents indépendants d'interconnectes et leurs donner la possibilité de coopérer selon des contraintes précises afin de résoudre les problèmes au sein du processus industriel via vise l'intégration de ces sous-systèmes grâce à une description de l'équipement unique et cohérente (à travers une description d'ontologie de l'équipement), une architecture générique (basée sur la technologie "Web Services") et des modèles cohérents de composants hétérogènes et des solutions technologiques

<sup>4</sup> ESPRIT : European Strategic Program on Research in Information Technology

### PROTEUS

Une plateforme européen fournissant des services web ou l'objectif principale est le développement d'une plateforme logicielle pour l'intégration et l'interopérabilité des modules logiciels de maintenance à distance à l'échelle industrielle « dont les fonctions sont la détection des alarmes, la gestion des pièces de rechange, l'optimisation des coûts et amélioration de la productivité » via l'intégration de ces sous-systèmes grâce à une description de l'équipement unique et cohérente (à travers une description d'ontologie de l'équipement). Une architecture générique (basée sur la technologie "Web Services") et des modèles cohérents de composants hétérogènes et des solutions technologiques.

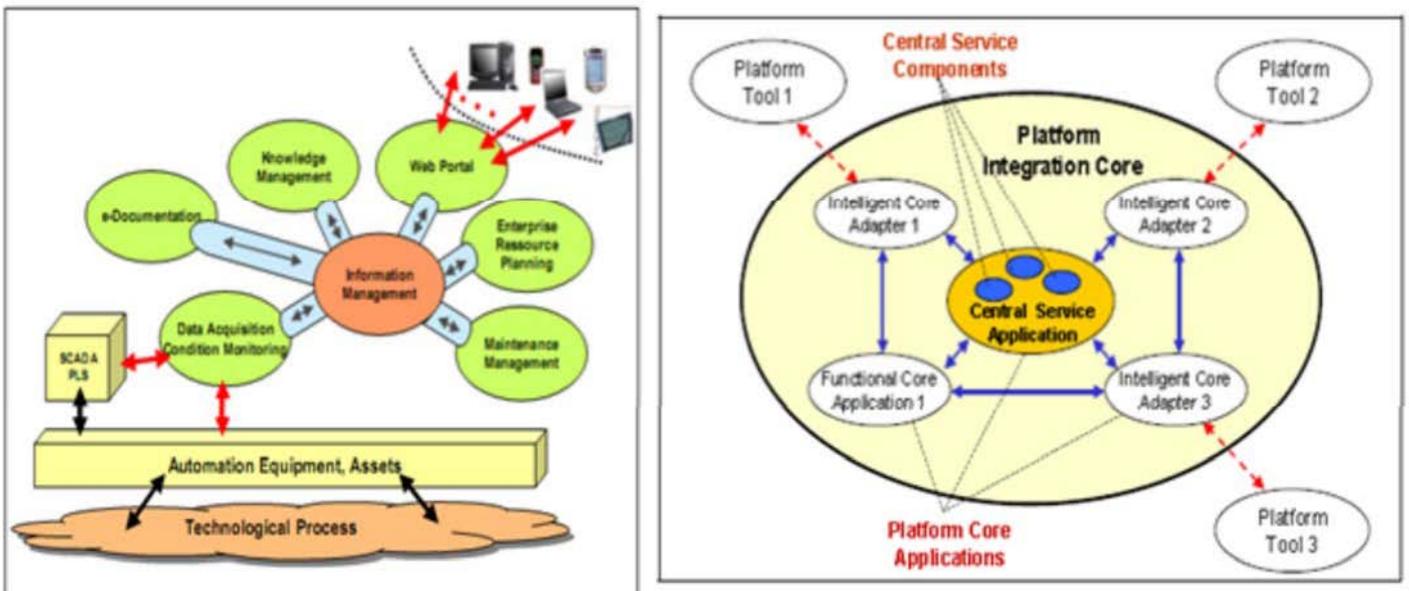


Figure 7: Architecture de la plateforme PROTEUS

Ces techniques permettent de garantir l'interopérabilité de systèmes hétérogènes afin d'assurer l'échange et le partage des informations, des données ainsi que des connaissances.

### PROMISE

PROMISE () un projet européen a pour objectif de la gestion des flux d'informations au-delà du client, de clôturer la boucle d'information de gestion du cycle de vie du produit (PLC, *Product Life Cycle*) et de transformer les données brutes du PLC<sup>5</sup> à des connaissances exploitables.

L'architecture utilise un modèle peer-to-peer<sup>6</sup> pour l'échange d'informations, où tout dispositif implémentant l'interface de messagerie de PROMISE (PMI-PROMISE Messaging

<sup>5</sup>PLC : est la succession des stratégies utilisées par la direction de l'entreprise en tant que produit passe par son cycle de vie <sup>6</sup>Peerr-to-peer : est un modèle de réseau informatique proche du modèle client-serveur mais où chaque client est aussi un serveur.

Interface) sur les services Web et peut communiquer avec tout autre dispositif prise en charge par le PMI, cette interface permet une approche sur les web-service.

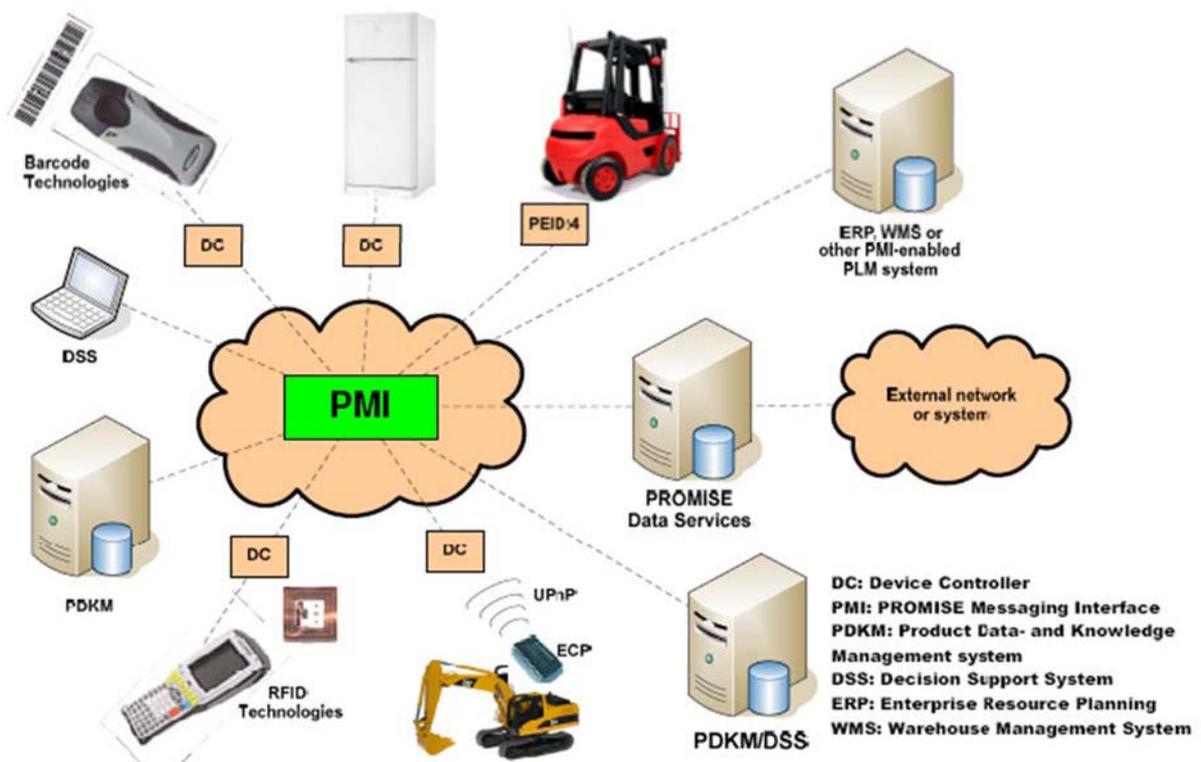


Figure 8: Architecture de la plateforme PROMISE

### SMMART

Le projet européen SMMART (System for Mobile Maintenance and Accessible in Real-Time) est système de maintenance mobile et accessible en temps réel, le système surveille l'utilisation et les données de la maintenance à travers le cycle de vie des pièces maitresses des équipements. Le système fournit aussi des outils de planification des ressources, outils avancés de dépannage, un contrôle de configuration et un service de traçabilité.

L'architecture proposée dans ce projet est basée sur la technologie RFID<sup>7</sup>-servant à la traçabilité des unités logistiques et dans le gestionnaire du moteur de configuration qui sert à la récupération les données de configurations des composants critiques.

<sup>7</sup>RFID : la technologie Radio frequency identification et Near Field Communication pour effectuer des paiements sécurisés

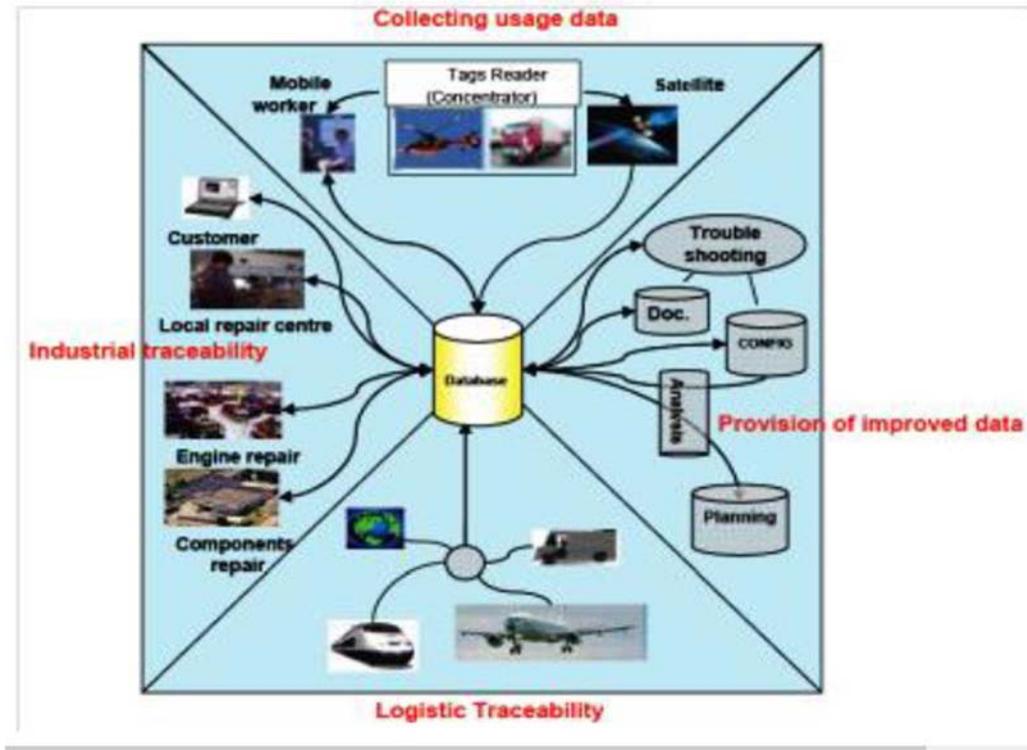


Figure 9: Architecture de la plateforme SMMART

### 3.3.9 Plateforme académique

Plusieurs plateformes académiques instanciées dans le cadre universitaire ou la plupart de ces plateformes restent proposition théorique sans réalisation concrète à l'exception de D2BTM.

#### **WSDF**

Cette plateforme entre dans le *cadre d'e-diagnostics basé sur des services Web*<sup>8</sup> (WSDF) visant à améliorer les systèmes de diagnostic à distance en fournissant automatiquement le mécanisme d'intégration des informations de diagnostic par internet. La plateforme WSDF a comme objectif soutenir les tâches d'e-diagnostics. Les technologies utilisées dans cette plateforme sont les Services Web, XML Signature, chiffrement XML, https, SOAP, ISDN... .

#### **D2BTM**

L'un des premiers projets académiques en e-maintenance a été élaboré au centre de systèmes intelligents de maintenance (IMS) aux Etats-Unis. Ce projet supporte le déploiement et l'expérimentation de la plateforme « dispositif-to-business » (D2BTM) prenant appui sur un élément de base qui est le Watchdog Agent<sup>9</sup>. L'objectif de cette plate-forme est de

<sup>8</sup>Web service : est un service offert par un dispositif électronique à un autre dispositif électronique, communiquant les uns avec les autres via le World Wide Web

<sup>9</sup> Agent qui Développe des outils et des algorithmes de pronostics innovants, des technologies de maintenance prédictives locales et à distance pour prédire et prévenir les défaillances des machines

transformer les données équipement à des données au format compatible Web (par exemple XML) de sorte que de nombreuses applications Web peuvent être exécutées.

### Plateforme orientée Agents

Ce système multi agent d'e-maintenance basé sur la gestion des connaissances, il permet l'échange de connaissance et d'information entre des systèmes automatiques industriels. Toutes les sources d'information dans les systèmes d'automation industriels sont analysées et modélisées par des agents

L'architecture de la plateforme est composée principalement de cinq types d'agents :

- l'agent de connaissance [Knowledge Agent (KA)],
- l'agent de configuration [CA (Configuration Agent)],
- l'agent de diagnostic [DA (Diagnosis Agent)],
- l'agent de récupérations des données [FA (Field Agent)]
- l'agent de management [MA (Management Agent)].

Ainsi, chaque agent est composé de cinq modules à savoir

- le module de connaissances local (LKM) qui définit les méta-connaissances relatives à la connaissance du système d'automatisation qui est représentée par l'agent.
- le module de connaissance sociale (SKM) qui définit les associations communautaires de l'agent.
- le module de prise de décision DMM qui définit les algorithmes et les modes de prise de décision.
- le module de communication (CoM) définit les mécanismes de communication et des protocoles.
- le module d'interface (ItfM) qui définit les spécifications pour les interfaces et les adaptateurs.

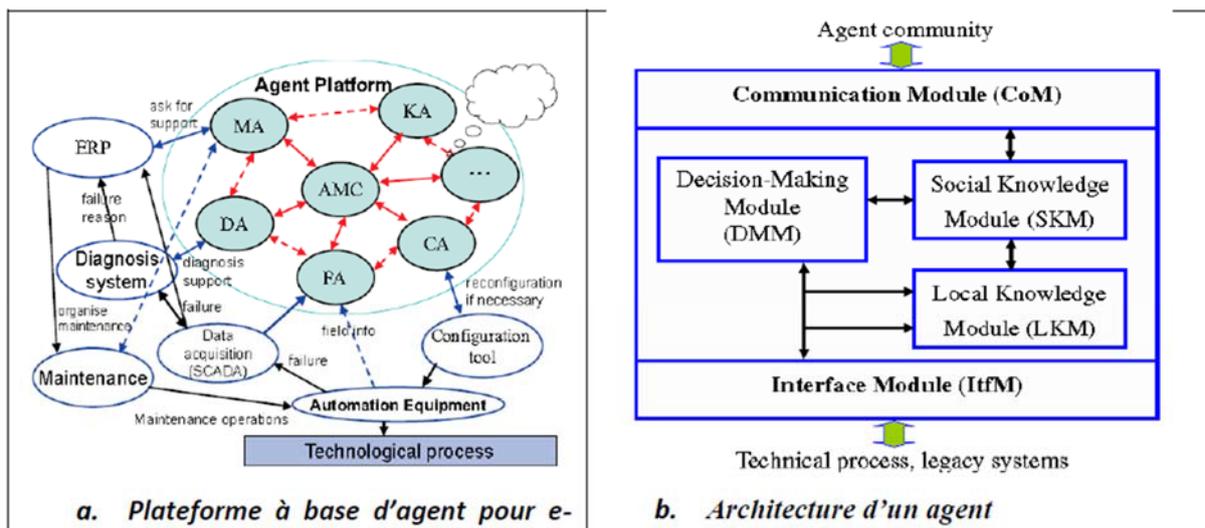


Figure 10: Architecture de la plateforme Orienté Agent

### ***IIEMD (Integrated Intelligent Equipment Maintenance Decision)***

Une plateforme composée d'un ensemble d'outils aide à la décision pour les différentes activités de la maintenance, soutenue par une ontologie ou deux types de raisonnement le RBR<sup>10</sup> (*Rules Based Reasoning*) et le CBR<sup>11</sup> (*Case Base Reasoning*). L'architecture de la plateforme IIEMD est basée sur la technologie Web, est modulaire, offrant la possibilité d'une intégration permansive pour plus d'agilité en temps réel.

Cette plateforme peut réaliser la présentation et la synthèse des données d'une manière compréhensive, l'évaluation continue de la santé des équipements, la prise de décisions sur les opérations de maintenance et de réparation, ainsi que l'intégration d'autres systèmes.

Dans cette architecture, les outils de décision de maintenance (tous sont définis comme des services Web) peuvent enregistrer, récupérer l'information à partir d'autres systèmes, distribuer les informations aux autres systèmes intégrés dans la plateforme. Ainsi, elle fournit un processus de décision automatisé, transparent, continu de façon unifiée, indépendamment de son origine, du fabricant, des intégrateurs et des utilisateurs finaux de l'équipement grâce des méthodes de raisonnements à base de cas et à base de règles associées à des règles d'adaptation.

#### **3.3.10 Les limites de l'e-maintenance**

Le but de la e-maintenance est d'intégrer différents systèmes et composant dans une seul plate-forme par l'échange de données entre ces derniers mais cela évoque un problème, les différents composants et applications produisent de différents format de données qui ne sont pas toujours compatibles. Ce qui induit des problèmes d'interopérabilités, de coopération et de partage de connaissances. L'interopérabilité est «la capacité de deux ou plusieurs systèmes ou composants à échanger des informations et à utiliser les informations qui ont été échangées»

### **3.4 S-Maintenance**

Dans le but d'améliorer les performances des processus de la maintenance fournis par la e-maintenance un nouveau concept est proposé celui de la s-maintenance. la s-maintenance tend vers « sémantique maintenance » Le principe du concept est de donner une sémantique aux informations et aux données partagées et échangées entre différent services et composant d'un système de maintenance moderne.

#### **3.4.1 Définition**

La s-maintenance est la réalisation de la maintenance basée sur la connaissance experte du domaine, où les systèmes dans le réseau gèrent ces connaissances et partagent la sémantique

---

<sup>10</sup> RBR : sont utilisés comme un moyen de stocker et de manipuler des connaissances pour interpréter l'information d'une manière utile. Ils sont souvent utilisés dans les applications d'intelligence artificielle et de la recherche.

<sup>11</sup> CBR : est le processus de résolution de nouveaux problèmes sur la base des solutions de problèmes similaires

faisant émerger de nouvelles génération de services et offrant des services à la demande, grâce à des fonctionnalités adaptatives et autonomes.

### 3.4.2 Définition de la plateforme de s-maintenance

Une plateforme de s-maintenance est un système collaboratif et distribué basé sur l'ingénierie des connaissances<sup>12</sup> fournissant des services dynamiques et des services à la demande selon les exigences de ses utilisateurs grâce à des fonctionnalités d'autogestion des processus de maintenance et d'auto-apprentissage »

### 3.4.3 Architecture de la s-maintenance :

L'architecture de la plate-forme de s-maintenance, plus performante au niveau de la communication et l'échange de données, prend appui sur l'architecture de la plate-forme d'e-maintenance avec un échange d'informations basé sur le web sémantique. La sémantique de l'information échangée nécessite la création d'ontologie du domaine commun aux différents systèmes, qui composent la plate-forme.

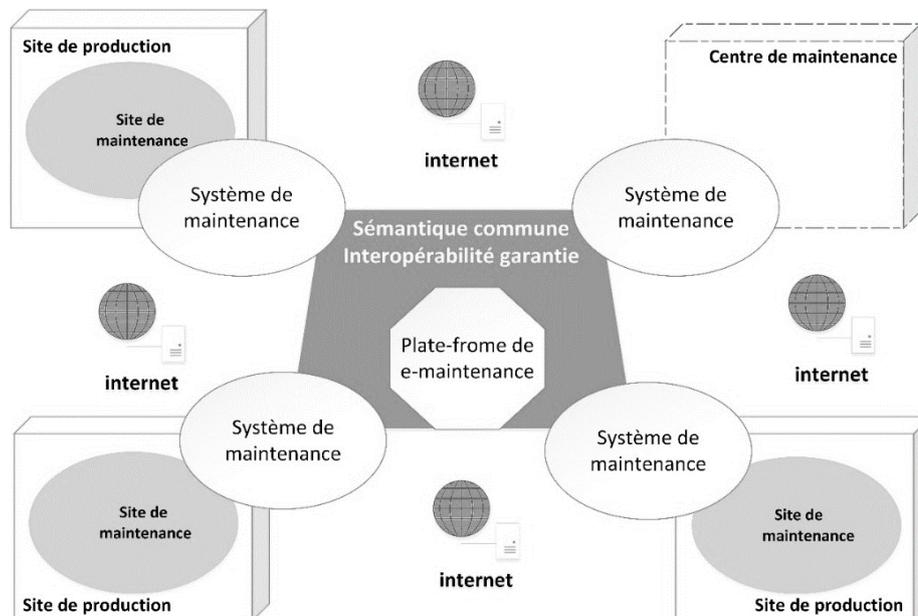


Figure 11: Architecture du concept de s-maintenance

### 3.4.4 Système Existant

Selon (Karray, 2012) les plateformes IIEMD et Plateforme Orientée Agents (POA) sont plus ou moins les plus proches de la plateforme de s-maintenance avec quelque réserve qui les empêchent d'être classées en tant que des plateformes s-maintenance.

La Plateforme Orientée Agents possède quelques points de concordance avec la s-maintenance au niveau de l'interopérabilité sémantique et l'exploitation de l'ingénierie des connaissances. Néanmoins, cette exploitation de connaissances est faite localement par un

<sup>12</sup>L'ingénierie des connaissances : Se concentre sur l'identification, la création, le stockage et la mise en disposition des connaissances.

agent et n'est partagées et réutilisées par toutes les applications d'aide à la maintenance intégrées dans la plateforme. De plus, les agents ne fournissent pas de services dynamiques.

La plateforme IEMD est l'une des plus proches de la s-maintenance. Elle se basant sur l'ingénierie des connaissances (ontologie et moteur d'inférence) et en met l'accent sur l'aspect coopératif. Toutefois elle ne s'intéresse pas aux services dynamiques et autonomes. De plus, IEMD ne traite que les services d'aides à la décision pour la réparation qui est une approche orientée utilisateur final (le technicien), mais elle ne développe pas sur la notion de génération de services à la demande pour les différents types d'utilisateurs de la plateforme (tous les acteurs de la maintenance).

### 3.4.4.1 Le Système Proposé

Une architecture d'un système a été proposée lors de la **9eme conférence internationale de l'informatique industrielle IEEE par MOHAMED KARRAY**, Basé sur les composants de la maintenance les caractéristiques de la s-maintenance. Le système proposé avec deux type de composants, composant générique et d'autre spécifique sont décrit ci-dessous ainsi que leurs fonctionnalités.

#### Composant Générique

- Un système d'acquisition de données
- Un système de GMAO
- Un progiciel de gestion intégrée (PGI)
- Un système de documentation e-Doc
- Un système de Diagnostic
- Un système de Pronostic
- Un système de planification
- Un Système de géolocalisation

#### Composants spécifiques

- a. Coordinator (Coordinateur)** : désormais la conduite principale de communication entre les utilisateurs avec le système et les applications avec le noyau. Il coordonne entre tous les composants et les applications intégrées du système
- b. Mediator (Médiateur)** : le rôle de ce composant est d'assurer l'interopérabilité sémantique entre les applications. L'architecture du système de s-maintenance est basée sur la connaissance et la sémantique représentée par une ontologie partagée entre ses systèmes intégrés. Les ontologies locales de ces systèmes se développent indépendamment les unes des autres, le médiateur prend en compte l'évolution des ontologies locales de ces systèmes intégrés
- c. HIM (Human Interface Manager/ gestionnaire d'interface humaine)**

Il gère tous les affichages sur l'interface Web en bonne forme et un contenu de qualité. Chaque acteur a accès au système de maintenance à travers HIM. Un personnel intervenant de la maintenance doit être capable de se connecter au système via un navigateur Web sur tout type de terminal client. Le terminal client peut être un PC traditionnel, téléphone mobile, ou toute appaillage auquel il est possible de se connecter à Internet.

### d. **AC** (Access Control/contrôle d'accès)

Différents acteurs peuvent intervenir sur le système. L'AC doit gérer les droits des utilisateurs ainsi que des applications. Une base de données d'authentification est utilisée pour limiter l'accès non autorisé. Le AC donne l'autorisation d'accès à un composant ou application demandée par une requête émise par un utilisateur ou une application.

### e. **KBM** (Knowledge Base Manager/gestionnaire de base de connaissance)

Gestion de la base de connaissance il vérifie la cohérence des sorties et entrées de la connaissance avec la base de l'ontologie<sup>13</sup> (tous de la connaissance que les autres composants se renseignent ou vous inscrire dans le KB, sont compatibles avec l'ontologie).

### f. **KB** (Knowledge Base/base de connaissance)

La base de connaissances<sup>14</sup> contient l'ontologie : concepts et règles (méta-modèle ontologique), l'ontologie (instances de méta-modèle ontologique), les données (instances d'ontologies)

### g. **Reasoner** (reasoning engine/ moteur de raisonnement)

Le moteur de raisonnement contient des algorithmes généraux qui peuvent se permettre de manipuler les connaissances stockées dans la base de connaissances. Son objectif global est de rechercher des informations et des relations de la base de connaissances pour fournir des réponses, des suggestions et des prévisions. De nouvelles connaissances sont générées à partir des connaissances existantes par le moteur de raisonnement.

### h. **SG** (Generator Service/générateur de service)

Quand un service qui n'existe pas est demandé, le coordonnateur demande au SG d'établir le nouveau service (des exigences, fournir des informations, le raisonnement, etc). Après la génération de ce nouveau service, il est inscrit dans le SLIB. Il peut être une composition de services de base définis dans le SLIB. Actuellement, un mécanisme est en cours de recherche pour mettre en œuvre ce composant.

### i. **Slib** (Service Library/bibliothèque de service)

Il contient tous les services qui ne sont pas fournis par les applications intégrés dans la plateforme et les services générés par le SG, tels que : la traçabilité de mode de fonctionnement de l'équipement ou l'analyse financière par des indicateurs.

### j. **PM** (Process Manager/gestionnaire de processus)

Le composant PM est chargé de gérer l'ensemble des processus qui passent à travers le système.

### k. **TM** (Traceability Manager/gestionnaire de traçabilité)

---

<sup>13</sup> Ontologie est l'ensemble structuré des termes et des concepts représentant le sens d'un champ d'information

<sup>14</sup> Une base de connaissance regroupe des données spécifiques à un domaine spécialisé, sous forme exploitable.

## Chapitre I: La Maintenance

---

Il permet de connaître les actions effectuées par le système. Alors que toutes les communications internes entre la plate-forme et les applications passent par le coordonnateur, le TM connecté au coordonnateur permet de retracer les interactions de la plateforme globale : les interactions entre les applications, les utilisateurs avec la plate-forme et les interactions entre les composants de la plate-forme.

### Fonctionnement Du Système

Il existe de sorte de d'interaction entre utilisateur et le système de s-maintenance le premier la demande d'authentification et le second est la demande de services, les services sont soit intégrés ou générés par le système.

Quand un utilisateur effectue une demande d'accès par l'interface web, sa demande passe via *HIM* vers le *Coordonateur*, ce dernier envoie la demande d'authentification au composant *AC* pour vérifier les droits d'accès de l'utilisateur à la plateforme. Après un accès réussi, l'utilisateur peut demander des services selon son droit d'accès.

Après le système traite deux cas possible après que le *Coordonateur* reçoit une demande de service :

Le premier est quand le *coordonateur* reçoit une demande de service fourni par les applications intégrées. Si c'est le cas le coordonnateur communique avec l'application qui fournit le service, l'application est lancée et génère le service demandé.

Le deuxième cas est quand le *coordonateur* reçoit un service qu'il n'est pas fourni par les applications intégrées. Dans ce cas le *coordonateur* cherche dans *Slib* (bibliothèque des services) pour vérifier l'existence du service. Si le *Slib* contient le service ce dernier est lancé à l'utilisateur, sinon le *coordonateur* demande au *SG* (générateur de service) de générer le service à l'aide des connaissances existantes.

### Fonctionnalité

Un service d'autogestion de la maintenance a été pris pour exemple pour présenter pour illustrer l'exécution du système en ce service vise à assurer le fonctionnement automatique sans l'intervention d'opérateur humain.

L'exemple traite RUL(Remaining Useful Life/durée de vie résiduelle) générée par une application de pronostique concernant un composant X d'un équipement Y. L'application de pronostique envoie une requête au *coordonateur* pour demander les données sur le composant X de l'équipement Y. le *coordonateur* demande ces données à SCADA (système de contrôle et d'acquisition de données) [est un système de surveillance et de contrôle des équipement.], ce dernier envoie les données demandées au *coordonateur* qu'il le transmet à l'application de pronostique. Après l'analyse l'application de pronostique constate que le composant X défailit dans T heures et transmet une requête au *coordonateur* qui demande au *PM* quelle action doit être faite dans le cadre du processus OSA-CBM. Le *PM* envoie au *coordonateur* qu'il va lancer un expertise. Le *Coordonateur* envoie une requête à l'application de pronostique pour établir un l'expertise sur l'équipement Y, cette dernier demande les données nécessaire pour l'expertise de la part du *coordonateur* qui apporte ces information

## Chapitre I: La Maintenance

de SCADA et du *KB* et les transmet à l'application du diagnostic qui demande aussi de la transmettre le modèle du comportement de l'équipement Y. Quand ce dernier n'est plus fourni par aucune application le *coordonateur* consulte la *Slib* pour si le modèle de comportement de cet équipement est disponible, s'il existe le *coordonateur* demande l'exécution de ce service, puis le *coordonateur* transmet le modèle généré par l'exécution de ce service à l'application de diagnostic. Après l'analyse et le raisonnement, l'application édite un rapport d'expertise recommandant remplacer le composant X et le composant Z qui sont en cause de la panne. Le *Coordonateur* demande au *PM* l'action suivante à l'expertise, le *PM* mentionne qu'un plan d'intervention va être mise en place, par conséquent le *coordonateur* demande à l'application de planification de planifier une intervention pour remplacer le composant X et le composant Z de l'équipement Y avant T heures. L'application de planification demande des pièces de rechange et des ressources humaines. Puis le *coordonateur* demande au *PM* l'action suivante qui recommande d'éditer l'ordre et l'enchaînement de l'intervention, le *coordonateur* demande à l'application e-doc d'éditer le document, enfin le *coordonateur* transmet le document de l'ordre de l'intervention à l'opérateur qui va intervenir pour la maintenance via *HIM*.

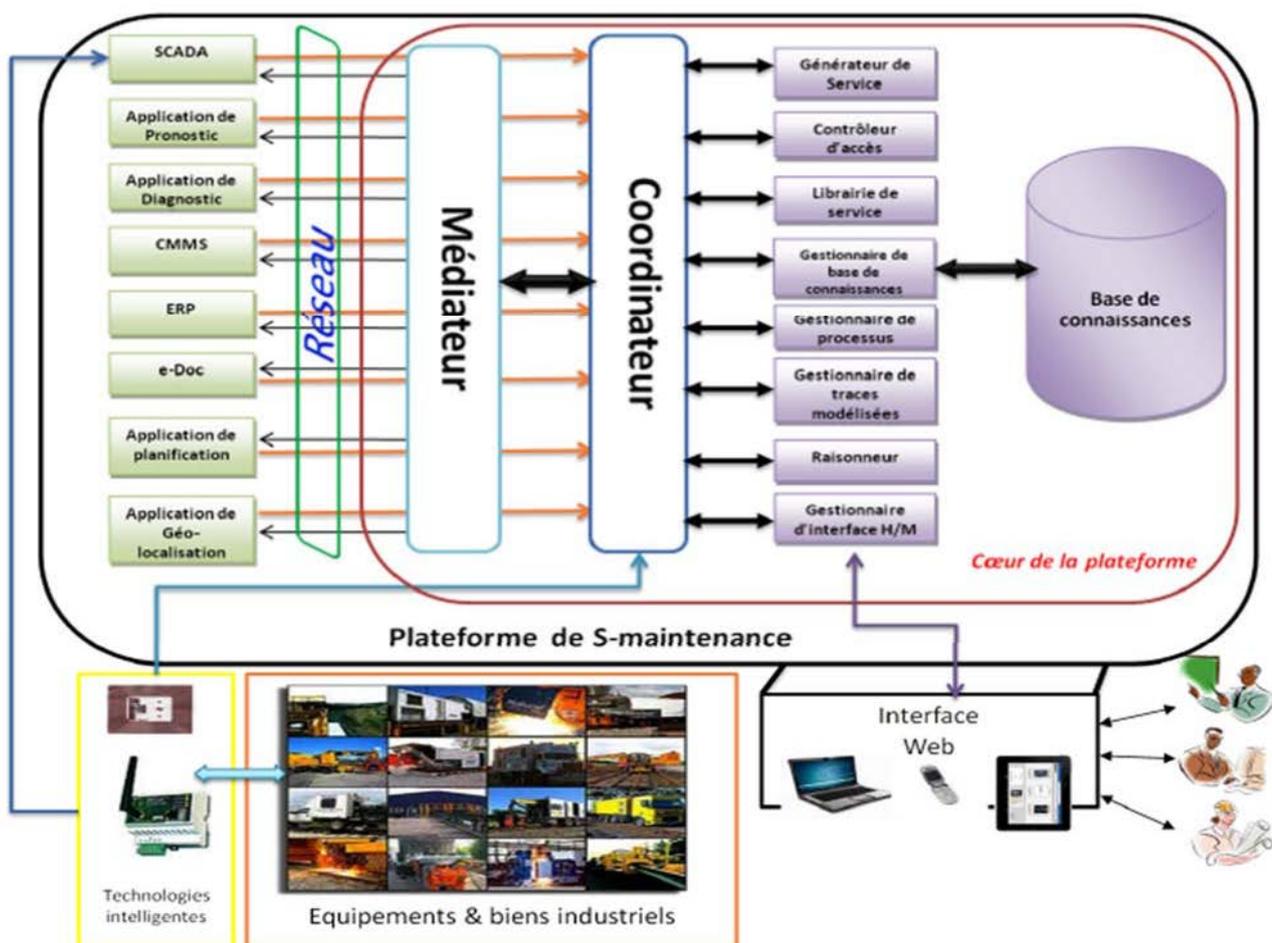


Figure 12: Architecture de la plateforme de S-maintenance proposé

### Conclusion :

D'après ce chapitre nous constatons que le système de maintenance a considérablement évolué dans le temps , il a commencé par un simple système de maintenance classique , puis il a évolué vers un système de maintenance assisté par l'ordinateur le GMAO, puis il a abouti à la e-maintenance qui est en quelque sorte plus intelligente en matière de communication ,entre les composants du début du système et sa fin .

Actuellement le système de la s-maintenance est en cours de développement et qui est une forme plus évoluée de l'e-maintenance basé essentiellement sur la sémantique.

Nous constatons aussi que l'informatique est toujours au cœur du développement des systèmes de maintenance.

# Chapitre II

---

Systeme Multi - Agents

### Introduction

Les systèmes multi-agents constituent aujourd'hui une nouvelle technologie pour la conception et le contrôle de systèmes complexes. Ils sont composés d'entités logicielles ou matérielles autonomes appelées agents, ces systèmes possèdent généralement plusieurs caractéristiques intéressantes, comme le parallélisme, la robustesse et l'extensibilité.

L'approche multi-agents repose sur plusieurs théories et concepts qui trouvent leurs origines dans plusieurs disciplines tels que la sociologie, la psychologie, les systèmes répartis, et le génie logiciel.

Dans ce chapitre, nous verrons les concepts fondamentaux de l'approche agent, nous insisterons sur l'évolution de l'aspect individuel (le comportement d'un agent seul) vers l'aspect collectif (Système multi agents) ; ensuite, nous étalerons une étude comparative sur les plates formes de développement des systèmes multi-agents ; avant de conclure avec des exemples d'application sur ce domaine.

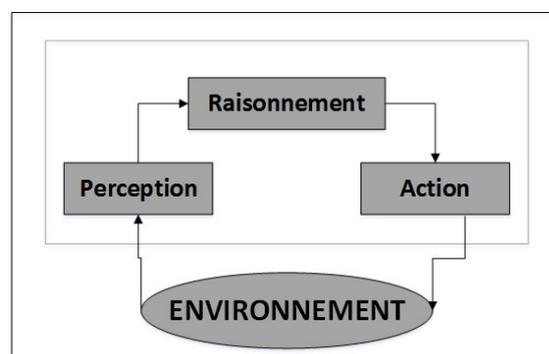
### 1) Les Agents

Durant ces dernières années, nous avons assisté à un fort et rapide développement des recherches sur les agents et les systèmes multi-agents. Le terme agent est un terme générique qui se rapporte à différentes entités (Franklin, 1996.)

- Des entités biologiques : les agents associés sont appelés agents biologiques.
- Des robots autonomes.
- Des logiciels informatiques et leurs composants qu'ils soient intégrés dans des systèmes d'exploitation ou des systèmes informatiques complexes.

#### 1.1 Définitions

Un agent est une entité physique ou virtuelle évoluant dans un environnement dont il n'a qu'une représentation partielle et sur lequel il peut agir. Il est capable de communiquer avec d'autres agents et est doté d'un comportement autonome." (Ferber, 1995)



*Figure 1: Fonctionnement d'un agent*

Cette définition aborde une notion essentielle : l'autonomie. En effet, ce concept est au centre de la problématique des agents. L'autonomie est la faculté d'avoir ou non le contrôle

## Chapitre II : les Systèmes Multi-Agents

de son comportement sans l'intervention d'autres agents ou d'êtres humains. Une autre notion importante abordée par cette définition concerne la capacité d'un agent à communiquer avec d'autres.

" Un agent est un système informatique capable d'agir de manière autonome et flexible dans un environnement. Par flexibilité on entend :

- **Réactivité** : un système réactif maintient un lien constant avec son environnement et répond aux changements qui y surviennent.
- **Proactivité** : un système proactif (aussi appelé téléonomique) génère et satisfait des buts. Son comportement n'est donc pas uniquement dirigé par des événements.
- **Capacités sociales** : un système social est capable d'interagir ou coopérer avec d'autres systèmes " (Wooldridge, 1999)

Donc un agent est une entité **physique** agit dans le monde réel. Un robot, un avion une voiture sont des exemples d'entités physiques. En revanche, un composant logiciel, un module informatique sont des entités **virtuelles**, car elles n'existent pas physiquement.

### 1.2 - Caractéristiques d'un agent

En partant de l'ouvrage de Wooldrige et Jennings, 1995, et des définitions citées, on peut identifier les caractéristiques suivantes pour la notion d'agent:

- **Situé** : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement.
- **Autonome**: l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.
- **Proactif** : l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment.
- **Capable de répondre à temps** : l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis.
- **Social** : l'agent doit être capable d'interagir avec des autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs. (Wooldrige. 1995)

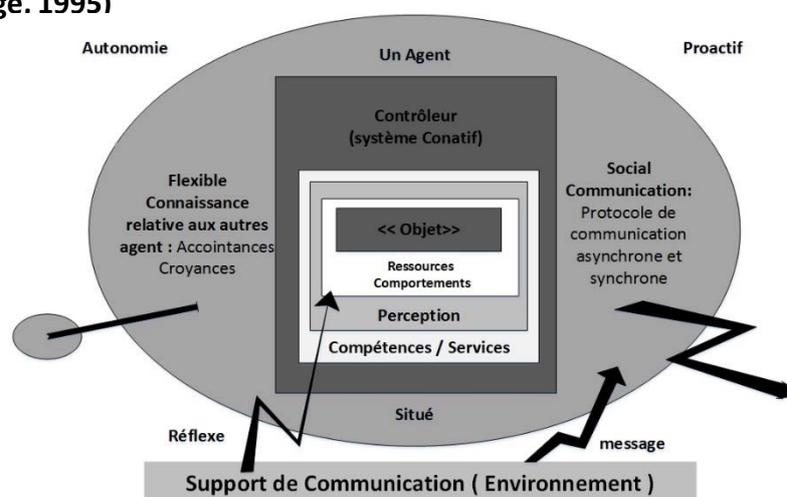


Figure 2: Propriétés d'un agent

### 1.3 - Catégories d'agents

Il existe une gradation dans l'intelligence d'un agent que l'on classe habituellement en Il existe trois catégories :

- **réactivité** : l'agent est uniquement dirigé par les événements perçus dans l'environnement. Il réagit de manière opportuniste à ces changements.
- **proactivité** : l'agent n'agit pas simplement en réponse à des changements de l'environnement, mais est aussi capable de s'assigner des buts et de prendre des initiatives pour les atteindre ;
- **sociabilité**: l'agent interagit avec les autres agents et les humains.

### 1.4 - Architectures d'agents

Suivant la façon dont un agent prend les décisions de ses réactions on peut distinguer plusieurs types d'agents.

#### 1.4.1 Agent basé sur la logique

Cette architecture est basée sur une représentation symbolique de l'environnement, ainsi que des actions que l'agent doit entreprendre. Puis une suite de déductions logiques qui permettent de faire la liaison entre les perceptions (qui sont déjà représentées symboliquement), et les actions qui conviennent.

#### 1.4.2 Agent délibératif

Un agent délibératif est un agent qui fait une suite de délibération pour choisir ses actions basant sur les buts.

#### 1.4.3 Agent BDI (Belief-Desire-Intention)

BDI<sup>1</sup> est l'acronyme de Belief-Desire-Intention qui signifie en français de croyances, désires, intention. Donc le comportement d'un agent BDI est basé sur ces trois concepts.

Un agent BDI est composé de :

1. **Un ensemble de croyances courantes** : représentent les informations de l'agent sur son environnement actuel.
2. **Une fonction de révision de croyances** : permet de mettre à jour les croyances de l'agent en fonction de ses croyances précédentes et de l'état actuel de l'environnement.
3. **Une fonction de génération des options** : elle détermine les options disponibles actuellement à l'agent (ses désires, ou encore les choix des actions à entreprendre).
4. **Un ensemble d'options courantes** : qui sont les désires ou les alternatives possibles pour l'agent.

---

<sup>1</sup> Acronyme pour *Beliefs - Desires - Intentions*.

5. **Une fonction de filtre** : c'est le processus de délibération de l'agent qui en fonction de croyances, désires, et intentions actuels de l'agent génère les nouvelles intentions.
6. **Un ensemble d'intentions courantes** : C'est le centre d'attention actuel de l'agent, c'est les objectifs qu'il essaye d'atteindre.
7. **Une fonction de sélection des actions** : cette fonction détermine les actions à effectuer en se basant sur les intentions actuelles de l'agent.

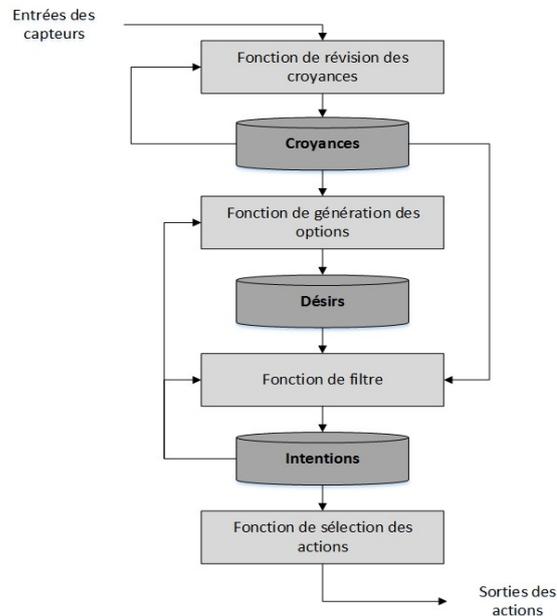


Figure 3 : L'architecture d'un agent BDI

### 1.4.4 Agents Cognitives et réactives

La distinction que l'on peut faire entre cognitif et réactif tient de la représentation du monde dont dispose l'agent. Si l'individu est doté d'une "représentation symbolique" du monde à partir de laquelle il est capable de formuler des raisonnements, on parlera d'agent cognitif tandis que s'il ne dispose que d'une "représentation Sub-symbolique", c'est-à-dire limitée à ses perceptions, on parlera d'agent réactif. La seconde distinction entre comportement téléonomique ou réflexe sépare les comportements intentionnels (poursuite de buts explicites<sup>2</sup>) des comportements liés à des perceptions. Les tendances des agents peuvent ainsi être exprimées explicitement dans les agents ou au contraire provenir de l'environnement. On peut construire un tableau regroupant les différents types d'agents (Ferber, 1995)

|                                  | <b>Agents cognitifs</b> | <b>Agents réactifs</b> |
|----------------------------------|-------------------------|------------------------|
| <b>Comportement téléonomique</b> | Agents intentionnels    | Agents pulsionnels     |
| Comportement réflexe             | Agents "modules"        | Agents tropiques       |

Tableau 1: Les différents types d'agent

<sup>2</sup> **Comportement téléonomique** : est celui qui doit sa finalité à l'opération d'un programme (Lille)

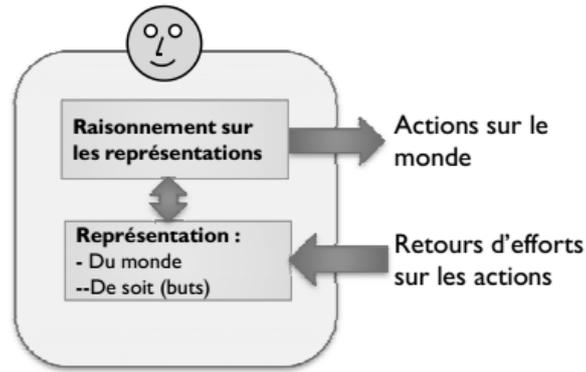


Figure 4 : Les agents cognitifs

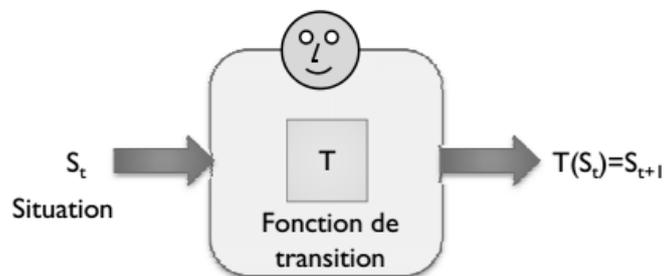


Figure 5 : Les agents réactifs

### 1.4.5 Architecture hybride ou architecture couches

Dans la majorité des cas un comportement purement réactif ou purement délibératif ne convient pas puisque on peut se retrouver dans des situations qui nécessitent une réponse et avec le même agent on peut se confronter à des situations que l'important n'est pas le temps mais la qualité de la réponse. (A.Chadli)

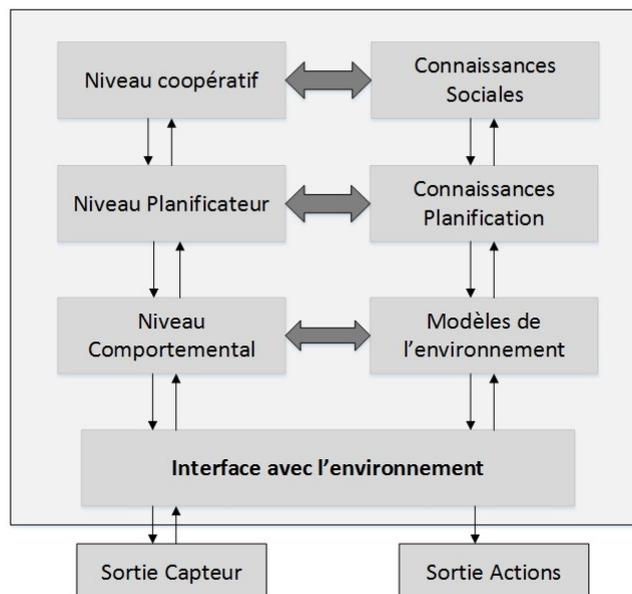


Figure 6 : L'architecture d'un agent Multi-Niveaux

### 2) Les Systèmes Multi-agents

Nous avons vu précédemment que les agents fonctionnaient dans des environnements qui peuvent être réels ou virtuels. Ces environnements peuvent permettre à des agents de se rencontrer et donner ainsi naissance à des communications ou encore des interactions. La communication consiste à envoyer des messages ; l'interaction, quant à elle, permet d'aller plus loin et d'établir des conversations structurées entre agents. C'est donc, comme le souligne Jacques Ferber (Ferber, 1995), une mise en relation dynamique de plusieurs agents par le biais d'actions réciproques qui influenceront le comportement futur des différents intervenants.

#### 2.1 Définitions

"On appelle système multi-agent (ou SMA), un système composé des éléments suivants:

1. **Un environnement E**, c'est-à-dire un espace disposant généralement d'une métrique.
  2. **Un ensemble d'objets O**. Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
  3. **Un ensemble A d'agents**, qui sont des objets particuliers (A inclut O), lesquels représentent les entités actives du système.
  4. **Un ensemble de relations R** qui unissent des objets (et donc des agents) entre eux.
  5. **Un ensemble d'opérations Op** permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.
  6. **Des opérateurs** chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers. "
- (Ferber, 1995)

Donc un système multi agents (SMA) est une sorte de société d'un ensemble d'agents qui coopèrent pour un but commun, qu'un agent ne peut pas le faire seul.

#### 2.2 Caractéristiques des systèmes multi-agents

- Chaque agent a des informations ou des capacités de résolution de problèmes limitées, ainsi chaque agent a un point de vue partiel.
  - Il n'y a aucun contrôle global du SMA.
  - Les données sont décentralisées.
  - Le calcul est asynchrone.
  - L'approche SMA ne s'intéresse pas à la recherche de solutions optimales. Elle recherche des solutions satisfaisantes pour des problèmes complexes.
- (A.Chadli)

### 2.3 La communication entre les agents

Les communications, dans les systèmes multi agents comme chez les êtres humains, sont à la base des interactions et de l'organisation. Une communication peut être définie comme une forme d'action locale d'un agent vers d'autres agents. Les questions abordées par un modèle de communication peuvent être résumées par l'interrogation suivante : qui communique quoi, à qui, quand, pourquoi, et comment ?

- **Pourquoi les agents communiquent-ils ?** La communication doit permettre la mise en œuvre de l'interaction et par conséquent la coopération et la coordination d'actions.
- **Quand les agents communiquent-ils ?** Les agents sont souvent confrontés à des situations où ils ont besoin d'interagir avec d'autres agents pour atteindre leurs buts locaux ou globaux. La difficulté réside dans l'identification de ces situations. Par exemple, une communication peut être sollicitée suite à une demande explicite par un autre agent.
- **Avec qui les agents communiquent-ils ?** les communications peuvent être sélectives sur un nombre restreint d'agents ou diffusées à l'ensemble des agents. Le choix de l'interlocuteur dépend essentiellement des accointances de l'agent (connaissances qu'a l'agent sur les autres agents).
- **Comment les agents communiquent-ils ?** La mise en œuvre de la communication nécessite un langage de communication compréhensible et commun à tous les agents. Il faut identifier les différents types de communication et définir les moyens permettant non seulement l'envoi et la réception de données mais aussi le transfert de connaissances avec une sémantique appropriée à chaque type de message.

### 2.4 Coopération

La coopération entre les agents consiste à décomposer les tâches en sous-tâches puis à les répartir entre les différents agents, il existe plusieurs décompositions possibles, le processus de décomposition doit donc tenir compte des ressources disponibles et des compétences des agents.

(SPINASSE, 2012)

Un agent doit coopérer avec d'autres agents du fait qu'il n'est pas capable de l'accomplir seul ou que les autres sont plus efficaces.

La coopération s'étend de :

- ☒ **la coopération totale** : Les agents peuvent délaissier leurs buts pour répondre aux besoins des autres agents afin d'assurer une meilleure **coordination**:
  - généralement mise en œuvre dans la résolution distribuée de problèmes
  - nécessite de nombreuses communications
- ☒ **à l'antagonisme total** : les agents ne coopèrent pas, leurs buts respectifs sont bloqués : les communications sont minimales => **négociation**

### Les agents ont :

- des buts compatibles
- des ressources suffisantes
- des capacités insuffisantes
- partage de tâches et d'informations

### 2.5 Interaction:

Une des principales propriétés de l'agent dans un SMA est celle d'interagir avec les autres agents. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent. Elles permettent aux agents de participer à la satisfaction d'un but global. Cette participation permet au système d'évoluer vers un de ses objectifs et d'avoir un comportement intelligent indépendamment du degré de complexité des agents qui le composent.

En général, les interactions sont mises en œuvre par un transfert d'informations entre agents ou entre les agents et leur environnement, soit par perception, ou bien par communication. Par la perception, les agents détiennent la connaissance de changement de comportement d'un tiers au travers du milieu. Par la communication, un agent fait un acte délibéré de transfert d'informations vers un ou plusieurs autres agents. (O.Boissier, 2001)

### 2.6 Plateformes multi agents

Le meilleur moyen pour construire un système multi-agents est d'utiliser une plateforme multi-agents. Une plate-forme multi-agent est un ensemble d'outils nécessaire à la construction et à la mise en service d'agents au sein d'un environnement spécifique. Ces outils peuvent servir également à l'analyse et au test du SMA ainsi créé. Ces outils peuvent être sous la forme d'environnement de programmation (API) et d'applications permettant d'aider le développeur.

Cet outil nous fournit un noyau de base pour la construction de notre prototype. Ce noyau doit nous fournir entre autres des services d'annuaires, le système de communication et des bibliothèques pour l'implémentation de comportement d'agents.

L'objectif de cette partie est de sélectionner une plate-forme qui correspond aux mieux à nos critères et déterminer si cet outil est bien adapté à nos besoins. Il existe deux sortes de plate-forme:

- **Les plates-formes académiques** : Plates-formes créées dans le cadre d'ateliers de recherche, souvent universitaires. Ces plates-formes sont distribuées gratuitement et leurs sources sont souvent en libre accès.
- **Les plates-formes commerciales** : Plates-formes créées par des entreprises et destinées à être commercialisées. On peut cependant obtenir des versions d'essais gratuites, mais les sources ne sont en aucun cas disponibles.

### 2.6.1 Critères de sélection

Pour choisir la plate-forme la mieux adaptée, nous nous sommes inspirés en effet sur deux types de critères :

- **Des critères généraux** proposés par Olivier Boissier, Zahia Guessoum et Michel Occello [Bgo 01]. Cependant, il a fallu faire une sélection parmi cette liste très importante, nous les avons résumés sous quatre critères, afin de mettre en avance les critères spécifiques pour orienter notre choix.
- **Des critères spécifiques** à nos besoins pour l'implémentation du système de maintenance basé sur les SMA. En fait, les fonctions essentielles seront axées sur la communication inter-agents et il suffira de vérifier que ces fonctions sont bien offertes par la plate-forme.

Autres des critères intra-agent de réutilisation, parallélisme et protocoles. Ces critères nous ont semblé importants pour faciliter l'implémentation des comportements de nos agents.

#### 2.6.1.1 Critères généraux

**Facilité d'apprentissage de l'outil et documentation de qualité** : Ce critère est déterminé en fonction de plusieurs facteurs dont la qualité de la documentation, la complexité des composants et les concepts utilisés. Les connaissances préalables à son utilisation comme le langage de programmation.

**Outils de « débogage »** : Plusieurs erreurs de coordination et de synchronisme risquent de se glisser à l'intérieur des programmes. La découverte (et la correction) de ces erreurs peut s'avérer très difficile voire même impossible sans outils appropriés.

**Diminution de l'effort demandé et Simplicité d'implémentation** : Dans ce critère, plusieurs facteurs doivent être pris en considération. Un langage supportant bien la programmation

orientée-objet, le « multi-threads » et la programmation réseau procurent des avantages importants. Les composants doivent aussi être facilement identifiables (nom, packages, documentation, paramètres, etc.). De plus, les classes et les services disponibles doivent être faciles à utiliser. La diminution de l'effort demandé en termes de quantité de code à écrire, de complexité des composants à implémenter, de facilité d'utilisation des composants existants sont aussi des facteurs à prendre en considération.

**Génération automatique de code** : Si les spécifications du système sont possibles au niveau des interfaces, il est important de pouvoir générer le code source du système (au moins les squelettes) et de ses différents composants.

### 2.6.1.2 Critères spécifiques

**Standardisation et gestion de l'interaction (Communication inter-agents satisfait-il la norme mise en place par la FIPA ?)** : nous ne devons pas avoir à se préoccuper de l'implémentation des connections entre les différentes machines, des protocoles de communications, de la sécurité, de la synchronisation, des services de messagerie et autres. Ces services doivent donc être déjà implémentés.

**Des agents prés définis du niveau méta (exemple les agents annuaires)** : il est possible de trouver des agents annuaires déjà implémenter sous forme de service ? Les agents contrôleurs doivent connaître les services des agents ressources. Chaque agent présent dans le système s'inscrit et publie ses compétences à travers les annuaires.

**Déploiement des agents contrôleurs** : la possibilité de répartir le système sur plusieurs machines est un critère très important au niveau de l'exécution. L'outil doit aussi permettre une exécution simple du système. L'exécution doit être indépendante de l'environnement. Ce critère est nécessaire pour satisfaire la notion de décentralisation.

**Support graphique pour la gestion des agents contrôleur** : l'environnement propose des interfaces graphiques facilitant la gestion des agents contrôleurs en exécution. L'outil permet l'interaction avec le système. Il permet par exemple, d'ajouter, de modifier ou de supprimer dynamiquement des agents dans le système. L'intérêt de ce type de gestion n'est pas à négliger, elle peut être très utile pour l'étude du pilotage au niveau de l'exécution, de la vérification et validation.

## Chapitre II : les Systèmes Multi-Agents

**Réutilisation et extensibilité du code des agents contrôleurs** : les utilitaires fournis par les outils comme les modules, les agents prédéfinis ou le code généré doivent être facilement modifiables.

Il faut aussi pouvoir ajouter facilement du code à celui existant.

**Parallélisme** : un autre point très important est la possibilité du parallélisme au niveau intra-agent. Un agent contrôleur a la capacité de jouer plusieurs rôles dans différents groupes donc le parallélisme des comportements d'agent contrôleurs est nécessaire.

### 2.6.2 L'évaluation

Le barème choisi est un code compris entre A et E qui est interprété comme suit :

- (A) si l'outil répond très bien au critère ;
- (B) si l'outil répond bien au critère ;
- (C) si l'outil répond moyennement au critère ;
- (D) si l'outil répond peu au critère ;
- (E) si l'outil ne répond pas du tout au critère ;

|                        |                                   | Plateforme |      |          |           |       |          |         |         |             |        |      |      |              |
|------------------------|-----------------------------------|------------|------|----------|-----------|-------|----------|---------|---------|-------------|--------|------|------|--------------|
|                        |                                   | Jade       | Zeus | Made kit | Agent oot | Decaf | Jat Lire | Ma leva | Fipa-os | Hive BeeGen | Jafmas | Bond | Jack | AgentBuilder |
| Critère                |                                   | Critère    |      |          |           |       |          |         |         |             |        |      |      |              |
| Critère<br>Spécifiques | Standardisation de l'interaction  | A          | C    | E        | C         | C     | B        | E       | A       | C           | B      | C    | C    | B            |
|                        | Déploiement                       | A          | D    | B        | C         | C     | C        | C       | A       | C           | D      | D    | C    | D            |
|                        | Réutilisation                     | B          | C    | B        | E         | D     | E        | C       | C       | D           | D      | C    | A    | D            |
|                        | Service d'annuaires               | A          | C    | E        | C         | E     | E        | E       | A       | E           | E      | E    | C    | C            |
|                        | Parallélisme                      | C          | D    | D        | C         | C     | C        | B       | C       | C           | C      | D    | C    | D            |
|                        | Support graphique pour la gestion | C          | B    | B        | B         | C     | C        | C       | C       | C           | C      | B    | B    | B            |
| Critère<br>Généraux    | Génération automatique de code    | E          | B    | C        | D         | E     | C        | C       | E       | C           | D      | B    | E    | D            |
|                        | Débugage                          | B          | B    | B        | C         | B     | C        | C       | C       | C           | C      | B    | E    | B            |
|                        | Apprentissage                     | C          | D    | E        | B         | B     | C        | C       | C       | C           | D      | D    | E    | D            |
|                        | Documentation                     | A          | A    | B        | C         | C     | C        | B       | A       | C           | C      | A    | B    | A            |

Tableau 2: Les résultats de l'évaluation

Notre objectif premier était de choisir une plate-forme multi-agents pour construire un prototype SMA pour notre système de maintenance. Nous avons choisi Jade, qui nous a paru, d'une part, très complet, notamment grâce à ses résultats dans le tableau précédent. **(H.SIABDELHADI)**

### 2.6.3 La plate-forme JADE :

JADE (Java Agent Development Framework - Bellifemine, Poggi, Rimassa, 1999) est une plate-forme multi-agents développée en Java par CSELT (Groupe de recherche de Gruppo Telecom, Italie) qui a comme but la construction des systèmes multi-agents et la réalisation d'applications conformes à la norme FIPA (FIPA, 1997). JADE comprend deux composantes de base : une plate-forme agents compatible FIPA et un paquet logiciel pour le développement des agents Java.

Le but de JADE est de simplifier le développement des systèmes multi-agents en conformité avec la norme FIPA pour réaliser des systèmes multi-agents interopérables. Pour atteindre ce but, JADE offre la liste suivante de **caractéristiques** au programmeur d'agents :

- **La plate-forme multi-agents compatible FIPA**, qui inclut le Système de Gestion d'Agents (AMS), le Facilitateur d'Annuaire (DF), et le Canal de Communication entre Agents (ACC). Ces trois agents sont automatiquement créés et activés quand la plate-forme est activée.
- **La plate-forme d'agents distribuée**. La plate-forme d'agents peut être distribuée sur plusieurs hôtes, à condition qu'il n'y ait pas de pare-feu entre ces hôtes. Une seule application Java, et donc une seule Machine Virtuelle Java, est exécutée sur chaque hôte. Les agents sont implémentés comme des threads d'exécution Java et les événements Java sont utilisés pour la communication efficace et légère entre agents sur un même hôte. Un agent peut exécuter des tâches parallèles et JADE planifie ces tâches d'une manière plus efficace (et même plus simple pour le programmeur) que la planification faite par la Machine Virtuelle Java pour les threads d'exécution.
- **Un certain nombre de DF** (Facilitateurs d'Annuaire) compatibles FIPA qui peuvent être activés quand on lance la plate-forme pour exécuter les applications multi-domaines, où la notion de domaine est la notion logique décrite par le document FIPA97 dans sa Partie 1.
- **Une interface de programmation** pour simplifier l'enregistrement de services d'agents avec un ou plusieurs domaines de type DF.
- Le mécanisme de transport et l'interface pour l'envoi et la réception des messages.
- **Le protocole IIOP** compatible avec le document FIPA97 pour connecter des plates-formes multi-agents différentes.
- **Le transport léger de messages ACL** sur la même plate-forme d'agents. Dans le but de simplifier la transmission, les messages internes (sur la même plateforme) sont transférés codés comme des objets Java et non comme des chaînes de caractères. Quand l'expéditeur ou

le récepteur n'appartient pas à la même plate-forme, le message est automatiquement converti à/du format de chaîne de caractères spécifiés par la FIPA. De cette façon, la conversion est cachée au programmeur d'agents, qui a seulement besoin de traiter la classe d'objets Java.

- **Une bibliothèque de protocoles** d'interaction compatibles FIPA.
- **L'enregistrement automatique d'agents** dans le Système de Gestion d'Agents (AMS).
- **Un service d'attribution de noms compatible FIPA** ; quand on lance la plate-forme, un agent obtient un identificateur unique (Globally Unique Identifier - GUID).
- **Une interface graphique utilisateur** pour gérer plusieurs agents et platesformes multi-agents en partant d'un agent unique. L'activité de chaque plateforme peut être supervisée et enregistrée.

**(RAHAB, 2011)**

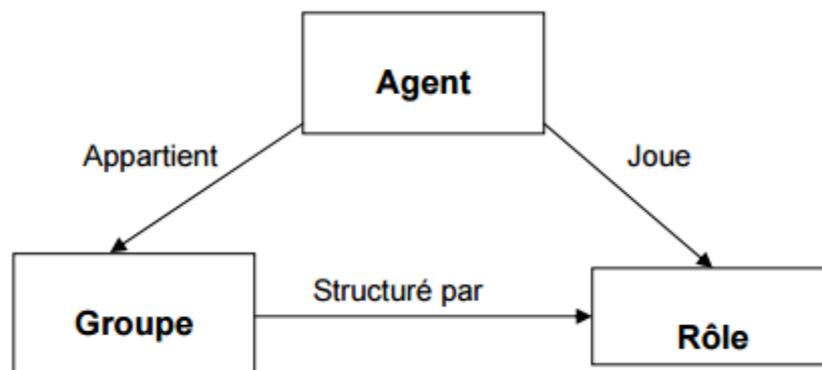
### 2.7 Exemples d'applications

- Applications industrielles: contrôle en temps réel, production, réseaux de télécommunications, systèmes de transport, systèmes de distribution, etc.
- Gestion de processus de business, support à la décision
- Commerce électronique
- Systèmes d'information coopératifs: découverte des sources, recherche de l'information, filtrage des informations, fusion des informations et personnalisation
- Interaction homme-machine
- Mondes virtuelles
- Divertissement  
(Florea)

### 2.8 Modèle organisationnel

#### 2.8.1 ALAADIN : Agent-Groupe-Rôle

On a tendance à qualifier d'organisation des structures sociales stables, ou bien présentant des normes sociales explicites. Nous utiliserons ce terme d'une manière plus générale, pour définir des sociétés d'agents analysables en ces termes, quitte à préciser le domaine d'étude et y raffiner les définitions selon le cas. Notre modèle est basé sur l'association de trois concepts clés : l'agent, le groupe et le rôle, utilisés simultanément pour décrire des organisations concrètes d'agents.



*Figure 7: Les trois concepts centraux*

On le résumera par la formule suivante : Un agent peut intervenir dans plusieurs communautés (que nous appellerons 'groupes' dans notre modèle) en parallèle. Il peut jouer dans chacun de ces groupes un ou plusieurs rôles correspondant à ses activités ou interactions. Ces rôles peuvent être portés par un nombre d'agents arbitraire, dépendant de la situation et des normes de l'organisation. (Gutknecht, 1998)

Voyons maintenant plus en détail ce que nous entendons ici par ces termes.

##### 2.8.1.1 L'Agent :

Quasiment aucune contrainte n'est posée sur l'architecture interne ou le modèle de comportement de l'agent. L'agent est simplement décrit comme une entité autonome communiquant qui joue des rôles au sein de différents groupes. La très faible sémantique associée à l'agent dans ce modèle est tout à fait volontaire. L'optique est bien de laisser la définition d'agent en retrait, non seulement pour ne pas prendre part dans le débat classique de "qu'est-ce qu'un agent", mais surtout pour laisser au concepteur toute liberté pour choisir l'architecture interne appropriée à son domaine applicatif.

### **2.8.1.2 Groupe :**

Le groupe est la notion primitive de regroupement d'agents. Chaque agent peut être membre d'un ou plusieurs groupes. D'une façon un peu simpliste, un groupe peut être vu comme un moyen d'identifier par regroupement un ensemble d'agents. Plus classiquement, associé au rôle, il définira la structuration organisationnelle d'un SMA usuel. Les différents groupes peuvent se recouper librement.

### **2.8.1.3 Rôle :**

Le rôle est une représentation abstraite d'une fonction, d'un service ou d'une identification d'un agent au sein d'un groupe particulier. Chaque agent peut avoir plusieurs rôles, un même rôle peut être tenu par plusieurs agents, et les rôles sont locaux aux groupes. L'hétérogénéité des situations d'interaction est rendue possible par le fait qu'un agent peut avoir plusieurs rôles distincts au sein de plusieurs groupes, les communications étant clôturées par les groupes.

### Conclusion

Les SMA, s'ils ne sont pas récents, sont actuellement un champ de recherche très actif, C'est une discipline qui s'intéresse aux comportements collectifs produits par les interactions de plusieurs entités appelées agents, ces interactions peuvent être mise en œuvre par des protocoles d'interactions.

Les systèmes multi agents peuvent être très utiles dans certaines situations et ils ont plusieurs avantages intéressants tels que la modularité, la vitesse et la fiabilité.

La technologie agent semble prendre de plus en plus d'importance ; En effet, elle permet de répondre aux besoins de nombreux domaines d'application jusque-là difficilement abordable par des moyens traditionnels ; Il s'agit notamment des systèmes complexes et des systèmes d'aide à la décision.

Plusieurs plates-formes multi-agents ont été abordé dans ce chapitre, on a présenté les principaux critères de chaque plateforme, notre choix s'est porté sur la plateforme JADE car son évaluation (voir tableau 2 ci-dessous) est positive dans la majorité des critères.

# Chapitre **III**

---

Modélisation et Conception

### Introduction

Ce chapitre traitera les étapes fondamentales concernant la conception et la modélisation du système, Ainsi nous allons décrire les approches, les outils et les concepts utilisés dans les conceptions et la modélisation du système en sujet.

Plusieurs approches et concept doivent être intégrés afin de réussir cette application ; en premier lieu l'approche d'Aalaadin concernant les environnements multi agent pour identifier les intervenants du système, en deuxième lieu nous allons entamer la fonction du pilotage des systèmes de production pour l'appliquer sur le système de maintenance.

Enfin nous aborderons le volet de la communication et la négociation entre les agents du système en expliquant l'utilisation des protocoles de FIPA ACL organisant les interactions entre les agents du système.

### 1. Le modèle Aalaadin:

#### 1.1 Présentation :

Le modèle Aalaadin décompose l'analyse des structures collectives en deux niveaux :

- **Le niveau méthodologique** : Définit l'ensemble des rôles possibles, spécifie les interactions et décrit les structures abstraites de groupe et d'organisation.
- **Le niveau descriptif** : Correspond aux concepts centraux d'agent, de groupe et de rôle. C'est à ce niveau que se décrit une organisation réelle. Nous considérons qu'une organisation des centres de décisions peut être décrite uniquement par ses concepts. Les relations entre agents sont donc primordiales par rapport à l'agent lui-même et à son comportement. Le groupe sera utilisé pour étudier la formation de différentes organisations. La notion du comportement de l'agent sera définie dans le chapitre suivant.

#### 1.2 Atouts du modèle Aaladin:

Les points forts du modèle d'ALAADIN peuvent être résumés comme suit :

**Hétérogénéité** : l'intervention d'un agent dans plusieurs groupes permet d'établir une "passerelle" d'information entre deux portions d'un système potentiellement incompatible. Dans ce cas l'agent interface capable d'interagir dans les deux modèles d'interaction relatifs à l'un et l'autre groupe (**Gutknecht Olivier, 2001**).

**Modularité** : Un agent agissant dans un groupe n'a pas le moyen de connaître l'activité d'un autre groupe auquel il n'appartient pas. Plus encore, à moins qu'un autre agent ne lui ait transmis une information spécifique, il n'en connaîtra même pas l'existence (**Gutknecht Olivier, 2001**).

**Fiabilité** : La présence d'agents au sein d'un système suppose l'adhésion de tels agents aux schémas d'interactions ou aux ontologies ayant cours dans ce système (**Gutknecht Olivier, 2001**).

### 1.3 Apports et limitations du modèle Aalaadin

Avec tous ces avantages ce modèle comporte un certain nombre de limitations :

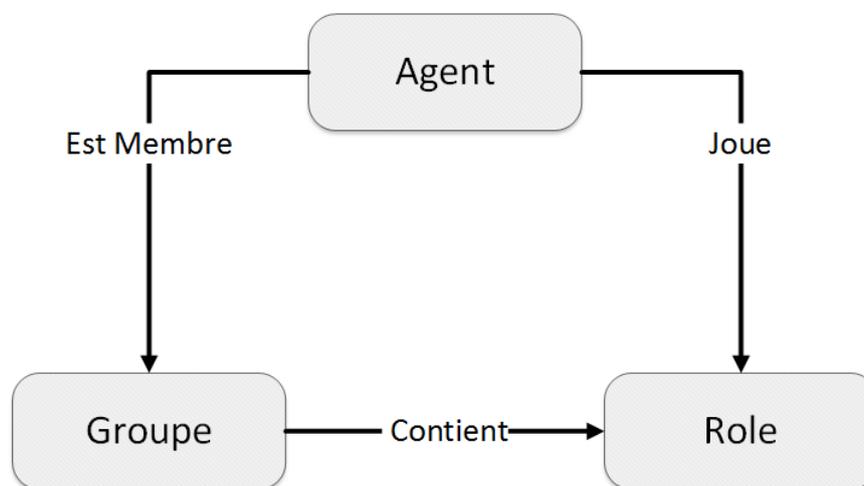
- L'environnement n'est pas mentionné dans cette approche alors que nous pensons que les interactions avec l'environnement forment une partie intégrante de l'organisation du système de pilotage. Dans la partie suivante nous verrons comment remédier à ce problème en s'appuyant sur un modèle d'interaction.
- Les schémas d'organisation sont soit attribués statiquement à un ensemble d'agents [DUR 96] ou disparaissent en tant que tel dans (Gutknecht, 1998). Une position intermédiaire est de permettre aux agents de manipuler explicitement des organisations (Muler, 2001). (Muller, 2002) et à terme, de faire émerger les organisations souhaitées (J-P, 1998)
- Il n'y a pas moyen de décrire comment une organisation peut être dynamiquement réorganisée dans une autre. Cette caractéristique est essentielle parce que l'exécution de différentes tâches peut demander de changer dynamiquement l'organisation pour chacune d'entre elles. Les transitions peuvent devenir même plus importantes que les états stables (J-P, 1998).
- Dans Alaadin, une organisation est vue comme un cadre (c'est-à-dire un Framework) pour les activités et les interactions grâce à la définition de groupes, de rôles et de leurs relations. La notion d'organisation correspond donc à une relation structurelle entre les rôles définis au sein des groupes. La notion d'interaction n'est pas réifiée (elle n'est pas explicitement représentée ni manipulable dans le système) et la méthode ne fournit pas de principes permettant de faciliter la décomposition d'un système complexe comme dans les systèmes de pilotage des systèmes de maintenance.

En résumé, la description des systèmes multi-agents en termes d'organisations est une abstraction très utile pour traiter des interactions à un plus haut niveau et avec la possibilité de gérer les fonctionnalités multiples d'un agent mais cette approche doit encore être approfondie dans le but de l'appliquer dans le domaine du pilotage des systèmes de maintenance.

Dans la partie suivante nous nous intéresserons principalement au modèle d'interaction du système multi-agents dédié au pilotage, où les agents interagissent par messages asynchrones, eux même exprimés dans un langage de communication d'agent.

### 2 Le modèle agent groupe rôle :

C'est un modèle de description sur lequel s'appuie notre proposition. Il permet d'exprimer et d'analyser divers systèmes multi-agents en utilisant avant tout des concepts organisationnels.



*Figure 1:Notion agent groupe role Aalaadin*

#### 2.1 Agent :

Dans ce modèle, il n'y a aucune contrainte ou pré requis sur l'architecture interne de l'agent et il n'y a aucun formalisme ou modèle particulier pour en décrire le comportement. Un agent est simplement défini comme une entité autonome communicante qui joue des rôles au sein de différents groupes. Nous pouvons reprocher à cette définition son flou, mais nous insistons sur le fait qu'elle est intentionnellement générale pour permettre à chaque concepteur d'agent de choisir parmi les modèles classiques le plus adapté à son application. Chaque acteur du système de maintenance est représenté par un agent. On distingue trois types d'agent :

**Agent ressource (réparateur) :** Les agents ressources pilotent les ressources nécessaires pour la réparation (pièces outillages) et se charge des interventions et des opérations effectuées dans le système.

**Agent équipement :** Les agents ressources sont responsable de leur traitement ils définissent les équipements en panne nécessitant une intervention de l'agent ressource.

**Agent opérateur :** L'agent opérateur qui représente les humains qui interagissent avec les agents ressources et les agents équipements dans le but de coopération.

### 2.2 Le Groupe

Le groupe est défini comme la notion primitive de regroupement d'agents. Chaque agent peut être membre d'un ou de plusieurs groupes. Dans sa forme la plus simple, un groupe peut être vu comme un moyen d'identifier par regroupement un ensemble d'agents, d'une manière plus évoluée, associé au rôle, il définira la structuration organisationnelle d'un SMA usuel. Un point majeur de cette définition est que les différents groupes peuvent se recouper librement. Un groupe peut être fondé par n'importe quel agent.

Nous définissons la notion de groupe dans un système de pilotage par un ensemble d'agents contrôleurs qui doivent coopérer pour réaliser un objectif. A chaque groupe on associe un objectif. Les objectifs peuvent être fixés par un agent opérateur. Avec cette notion de groupe l'organisation est dynamique. Ce qui signifie que les systèmes de pilotage peuvent se doter de nouvelles finalités (exemple, si le système de pilotage doit assurer un nouveau objectif un groupe composé d'agents contrôleurs se crée en temps réel pour réaliser cet objectif). Les objectifs du pilotage sont liés à l'environnement industriel de l'entreprise.

A partir de la définition des systèmes de la maintenance nous pouvons identifier des groupes dont les éléments sont de même type que ceux utilisés pour réaliser des objectifs locaux.

#### Groupes homogènes :

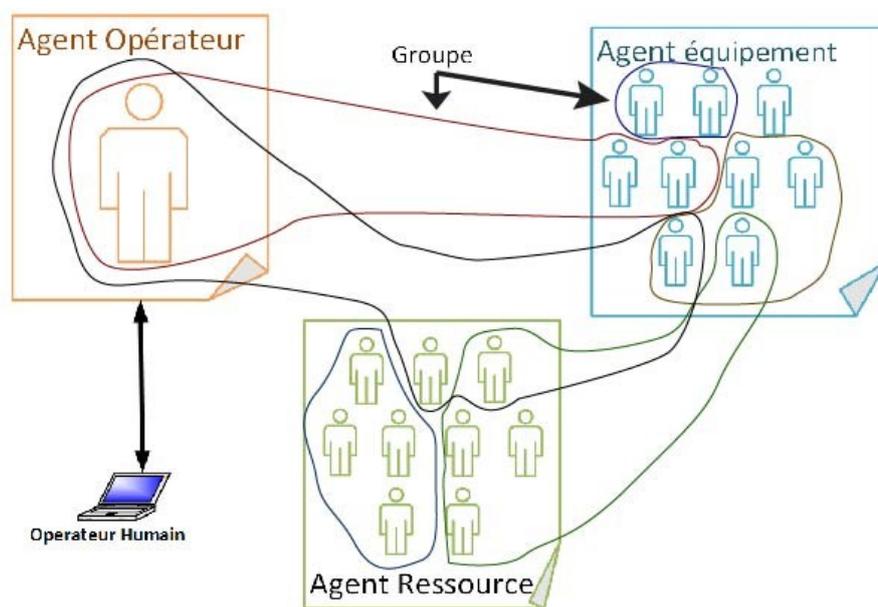
1. un groupe d'agents ressources (réparateur)
2. un groupe d'agents équipements
3. un groupe d'agents opérateurs (dans le but d'échanger les informations ou communications)

### Groupes hétérogènes :

Dont les éléments peuvent être de type différent par exemple :

1. Un groupe permettant d'associer un agent équipement avec les agents ressources nécessaires pour réaliser sa tâche.
2. Un groupe permettant d'associer un agent opérateur avec des agents ressources.
3. un groupe permettant d'associer un agent opérateur avec des agents équipements.

### 2.3 Le Rôle :



*Figure 2 : Notion de Groupe Aaladin*

Le rôle est une représentation abstraite d'une fonction, d'un service ou d'une identification d'un agent au sein d'un groupe particulier. Chaque agent peut avoir plusieurs rôles, un même rôle peut être tenu par plusieurs agents, et les rôles sont locaux aux groupes. La tenue d'un rôle dans un groupe préexistant doit être demandée par l'agent et n'est pas forcément accordée. La maîtrise de l'hétérogénéité des situations d'interaction est rendue possible par le fait qu'un agent peut avoir plusieurs rôles distincts au sein de plusieurs groupes, et que les interactions sont toujours locales à un groupe. Dans ce modèle il n'y a pas de définition de mécanisme particulier pour contrôler cet accès.

### 2.3.1 Rôle de l'agent d'Équipement :

Dans le système de maintenance nous pouvons trouver plusieurs agents équipements de même type qui sont créés dès leurs entrés dans le système et détruits à leurs sorties. Le rôle majeur de l'agent équipement est de simuler les équipements et illustrer les pannes qui peuvent se produire.

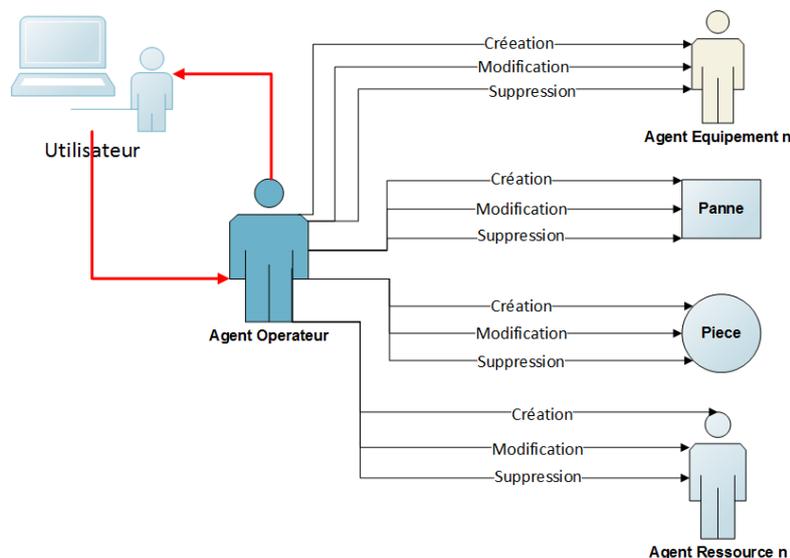
### 2.3.2 Rôle de l'agent ressource :

Ses agents pilotent les ressources (nécessaire pour le processus de maintenance du système de maintenance (exemples : les unités de traitements, les robots, les convoyeurs). Dans un système de maintenance on peut trouver des agents ressources du même type qui doivent coopérer entre eux pour éviter des conflits. Le rôle de l'agent ressource est d'effectuer le traitement et les tâches sur équipements en respectant les contraintes de temps, de coût et de qualité.

### 2.3.3 Rôle de l'agent Opérateur :

Nous définissons l'agent opérateur comme l'humain qui observe et aussi aide le système à s'auto organiser. L'opérateur interagit avec le système à travers une interface. Parmi les rôles de l'agent opérateur on trouve :

- a) L'introduction des nouvelles contraintes sur les ressources ou sur les pièces.
- b) D'influencer le système pour favoriser une tache par rapport à une autres.
- c) D'ajouter des nouvelles données et informations extérieures.
- d) De lancer et d'arrêter le fonctionnement du système.
- e) choisir le critère de l'appel d'offre



### 3. Le modèle d'interaction :

Pour un agent, interagir avec un autre constitue à la fois la source de sa puissance et l'origine de ses problèmes. C'est en effet parce qu'ils coopèrent que des agents peuvent accomplir plus que la somme de leurs actions, mais c'est aussi à cause de leur multitude qu'ils doivent coordonner leurs actions et résoudre des conflits.

Traiter le problème de l'interaction, c'est se donner les moyens non seulement de décrire les mécanismes élémentaires permettant aux agents d'interagir, mais aussi d'analyser et de concevoir différentes formes d'interaction que des agents peuvent pratiquer pour accomplir leurs tâches et satisfaire leurs buts.

Tout d'abord, les agents doivent être capables, par le biais de la communication, de transmettre des informations, mais surtout d'induire chez l'autre un comportement spécifique. Communiquer est donc une forme d'action particulière qui, au lieu de s'appliquer à la transformation de l'environnement, tend à une modification de l'état mental du destinataire. Par exemple, demander à un autre d'exécuter une tâche tend à provoquer chez l'autre une intention d'accomplir cette tâche et constitue donc une manière de satisfaire un objectif sans réaliser la tâche soi-même.

L'interaction consiste aussi à dire comment le système multi-agents va résoudre les problèmes de conflits, comment les entités doivent coopérer pour réaliser leurs buts et l'outil avec lequel les agents échangent des connaissances. Dans cette partie nous allons définir les modèles de communication et de négociation utilisés.

Dans ce qui suit nous allons expliquer le modèle conceptuel de base qui est le modèle temps réel pour le pilotage des systèmes industriels dont notre système en fait partie. A partir de ce modèle nous allons présenter notre vision de la résolution du problème de pilotage par le biais de l'interactivité.

### 4. Proposition d'un modèle de pilotage basé sur l'interactivité

Le pilotage est une fonction relativement complexe ayant donné lieu à quelques modèles conceptuels. L'intérêt de ces modèles est double. Ils permettent tout d'abord de - faciliter la compréhension des phénomènes mis en jeu, de - montrer les interactions du système

observé avec son environnement et - d'offrir un cadre pour la décomposition du système complexe en sous-systèmes plus simples. Le second intérêt est de pouvoir spécifier une architecture de pilotage afin de pouvoir ensuite les implanter.

### 4.1 Les fonctions du pilotage

Plusieurs travaux réalisés dans le domaine du pilotage (approche du N.I.S.T [JON 90], approche P.A.C.-Esprit 477-[HIG 90][LYO 90], approche du projet Esprit 809[VAN 90], approche P.C.S. (Planification, Conduite, Suivi) [ARC 91], l'approche composants [HUG 96]) et les normes Afnor ont convergé vers une représentation de l'activité de pilotage constituée de quatre grandes fonctions (planifier, exécuter, suivre, réagir) organisées comme indiqué sur la figure(3,4).

- La fonction **planifier** propose une solution d'ordonnancement de et fixe des objectifs pour le système de maintenance.
- La fonction **exécuter** transmet les ordres qui ont été planifiés au système en tenant compte de l'état des ressources,
- La fonction **suivre** contrôle le fonctionnement du système de maintenance et fournit des informations sur son état.
- La fonction **réagir** corrige les dérivations en respectant au mieux les objectifs. Elle a elle-même des capacités pour la planification (dans le cas ou des tâches qui ne nécessitent pas une intervention du module de planification), c'est-à-dire qu'elle peut adapter partiellement

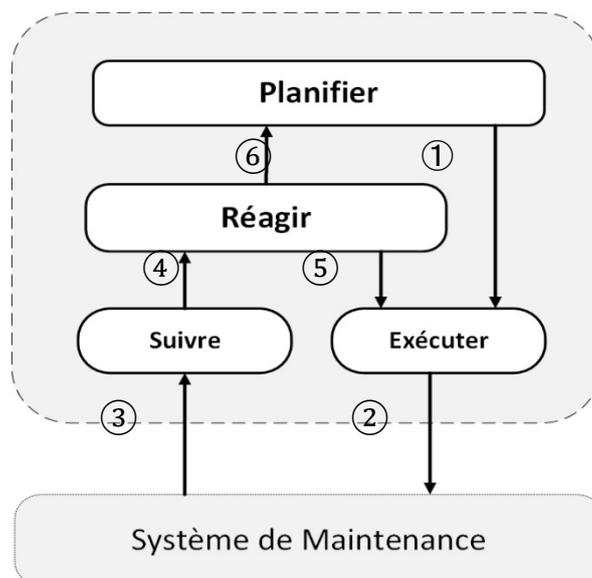


Figure 4: Fonction pilotage du système de maintenance

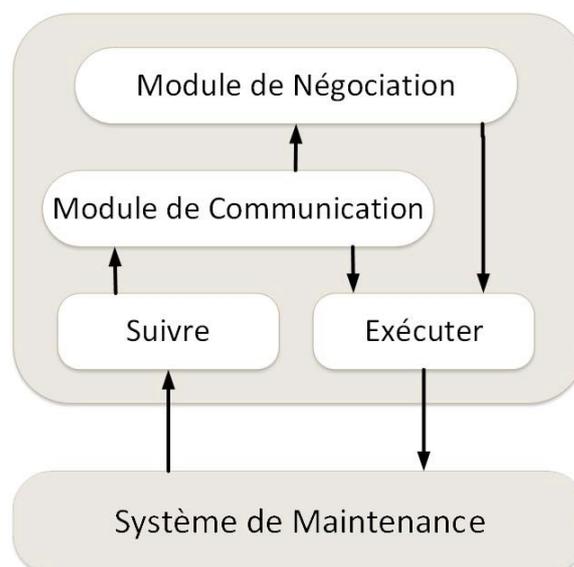
à un planning pour répondre à une perturbation.

La figure 3 illustre un modèle de pilotage :

① Les ordres et les instructions planifiées seront transmettre au module d'exécution qui les exécute dans le système ②. L'état du système est contrôlé et surveillé ③ par le module de suivi qui transmet les informations aux module de réalisation④, ce dernier aura deux possibilité selon les informations issus du module suivi ; soit : il réagit directement et prend l'initiative de transmettre des ordres au module d'exécution ⑤. Ou bien il retransmet ces informations au module de planification ⑥.

### 4.2 Architecture de pilotage

A partir du modèle conceptuel du pilotage des systèmes industriels (maintenance) et du modèle organisationnel issue du projet (Aalaadin), nous allons essayer de voir le modèle de pilotage d'une autre manière. Notre but est de traiter les activités complexes sous forme d'interactions entre agents plutôt que sur un raisonnement d'une seule entité qui planifie et décide. Pour cela nous allons transformer deux modules qui sont : planifier et réagir par la négociation et la communication. .



*Figure 5 : Pilotage du système de maintenance avec modules de communications et de négociations*

### 5. Protocole d'interaction :

Pour satisfaire un but les agents peuvent avoir des conversations en envoyant des performatifs entre eux. L'ordre des performatifs dans une conversation définit un protocole d'interaction. Il y a quelques modèles typiques dans les protocoles d'interaction, et un ordre spécifique des performatifs est prévu dans chaque modèle. De tels modèles typiques sont indiqués comme le protocole de FIPA. Nous présenterons deux des protocoles FIPA. Nous pouvons voir un protocole d'interaction comme une combinaison de plusieurs rôles compatibles, définissant les règles d'interaction pour la réalisation d'une tâche collective. Nous allons expliquer le lien entre le rôle et le protocole par la suite.

#### 5.1 Le protocole de communication :

Pour le protocole FIPA-Request, un agent envoie un message à un autre agent dans le but d'exécuter des actions et l'agent récepteur renvoie une réponse selon l'exécution ou non de l'action. Les trois diagrammes suivants sur la figure (3,7) illustrent la procédure du protocole FIPA-Request.

Supposons que l'agent maintenance invite l'agent ressource à exécuter l'action "action"

1. L'agent équipement envoie un message avec le performatif « request » à l'agent ressource. Si l'agent ressource est d'accord avec la demande, il envoie un message avec le performatif « agree ». Quand l'agent ressource finit d'exécuter l'action, il informe l'agent équipement en envoyant un message avec le performatif « inform ».
2. Si l'agent ressource ne peut pas exécuter l'action pour quelques raisons, il renvoie un message avec le performatif « failure » et les raisons.
3. Si l'agent ressource ne peut pas être d'accord avec la demande de l'agent Maintenance, il renvoie un message avec le performatif « refuse » et les raisons du refus

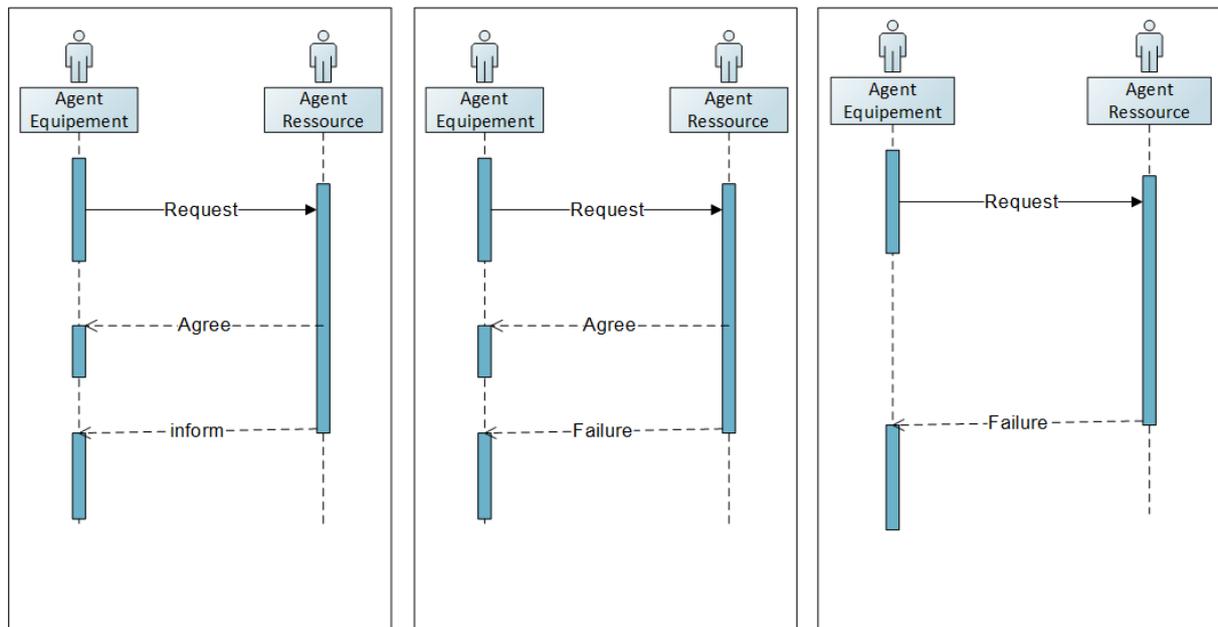


Figure 6 : communication sous protocole FIPA Request

### 5.2 Protocole de négociation :

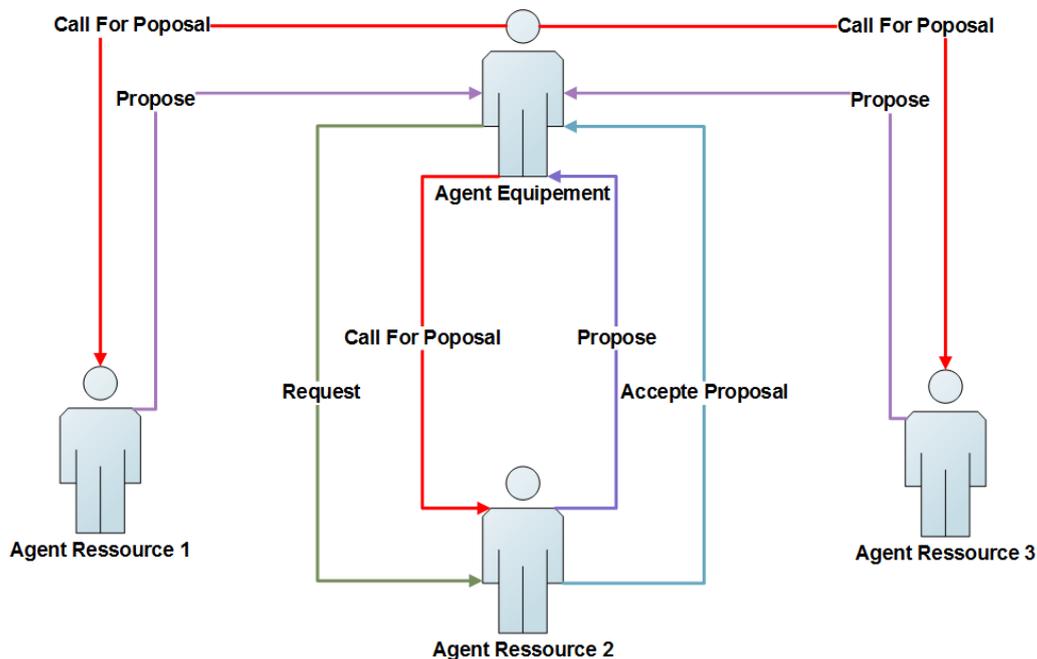
Nous avons choisi le Contract Net comme modèle de négociation. Il a été défini par Smith [SMI 80]. Le CNP (Contract Net Protocol) est un mécanisme de négociation par appel d'offre (ou Contract) entre deux types d'agent : l'agent gestionnaire (équipement) et les agents contractants (ressource). L'agent gestionnaire, souhaitant sous-traiter une tâche qu'il doit accomplir, est l'initiateur du contrat.

Chaque agent contractant est un agent auquel on propose ce contrat. Le fonctionnement du CNP a été décrit dans (Sassi M, 1992). Ce protocole très souvent utilisé, a été normalisé par l'organisation FIPA « fondation for intelligent physical Agent ». Nous avons adapté ce protocole pour le pilotage des systèmes de maintenance (Tchikou M, 2002).

Etant donné une ou plusieurs tâches (gamme), un gestionnaire (Agent maintenance), un groupe de (N-1) contractants (agents ressources):

- L'agent équipement envoie un message de type CFP « call for proposal » (décrivant les tâches (panne) à sous-traiter) à un groupe d'agents ressources (on diffuse l'annonce à tous les contractants).
- Chaque agent ressource évalue la tâche annoncée à l'aide d'une fonction locale évalue-annonce.

- L'évaluation précédente permet à un certain nombre d'agents de soumettre une proposition à l'aide du performatif « PROPOSE » à l'agent équipement.



*Figure 7 : Négociation sous protocole CNP*

- Si une proposition est jugée satisfaisante (à l'aide de la fonction évalue-soumission selon le critère choisis au préalable, alors la proposition est retenue. Il envoie également un message de type « REJECT\_PROSAL » (refus) aux agents ressources dont les propositions n'ont pas été retenues.
- L'agent ayant obtenu le contrat, remet un rapport d'exécution lorsque la tâche est complétée.
- Dans le cas de pannes ou de perturbations, l'agent équipement annule les contrats et re-planifie un nouveau plan en lançant un nouvel appel d'offre pour poursuivre les traitements.
- L'utilisateur peut également influencer le système en choisissant un contractant précis même si n'est pas retenu par le CFP.

Tableau : Les performatifs d'ACL utilisés pour la négociation

| Nom             | Type d'acte              | Signification   |
|-----------------|--------------------------|---|
| CANCEL          | Directive                | Annulation d'une requête  |
| PROPOSE         | Engagement<br>Directive  | Proposition avec éventuellement des conditions supplémentaires                                  |
| ACCEPT_PROPOSAL | Engagement               | Accepte une proposition et ses conditions   |
| REJECT_PROPOSAL | Engagement               | Rejette une proposition et ses conditions   |
| CFP             | Directive                | Appel à proposition pour une tâche  |
| FAILURE         | Assertion                | Informe de l'échec du locuteur  |
| NOT_UNDERSTOOD  | Déclaration<br>assertion | Incapacité du locuteur à interpréter le message ou non-respect du protocole par l'interlocuteur |

*Tableau 1: Les performatifs d'ACL utilisés pour la négociation*

### 6. Le lien entre le rôle et l'interaction dans le domaine du pilotage :

Dans ce paragraphe, Nous soulignons la nécessité de spécifier les rôles que les agents endosseront pour piloter un système de production [TCH 02]. Nous définissons le lien entre les deux notions : rôle et interaction. Une interaction est représentée comme un protocole entre différents rôles.

Dans notre modèle SMA la notion de rôle peut être vue sur plusieurs niveaux (Tchikou M, 2002).

#### a) - Les rôles élémentaires (niveau 1) :

C'est le rôle le plus simple qu'un agent puisse faire. Parmi les rôles élémentaires, on trouve celui d'initiateur d'une négociation, le rôle d'offrant et le rôle d'exécution. Il est important de souligner qu'un rôle élémentaire est intimement lié avec le protocole dont il fait partie, qu'il conditionne. La spécification d'un rôle fait nécessairement référence aux autres rôles élémentaires, avec lesquels il est censé interagir, même si elle ne spécifie pas leurs comportements complètement.

Deux rôles élémentaires sont compatibles lorsque, dans aucun cas de figure, un des rôles élémentaire ne se retrouve dans un état tel qu'un autre rôle élémentaire lui envoie un message (typé par un performatif) qu'il n'attendait pas ou ne lui envoie aucun message alors qu'il en attendait un.

### **b) - Les rôles complexes (niveau 2) :**

On distingue le rôle d'agent Equipement, le rôle de l'agent Ressource (réparateur) et le rôle de l'agent opérateur.

On peut définir le rôle complexe comme étant l'ensemble des responsabilités définissant le comportement attendu d'un agent dans le cadre d'une interaction avec d'autres agents, en d'autres termes c'est un ensemble de rôles élémentaires. Il est une spécification et une vue subjective, située, du comportement de l'agent dans le contexte de cette interaction. Un agent peut jouer plusieurs rôles élémentaires simultanément, dans une ou plusieurs conversations différentes. Par exemple, un agent Equipement peut négocier son traitement avec les ressources et gérer sa boîte aux lettres.

### **c) - Les rôles émergents (niveau 3) :**

Ces rôles sont propres aux groupes et émergent de l'interaction entre les agents. Par exemple effectuer le traitement sur l'équipement en respectant les contraintes de gamme, favoriser la réparation d'un équipement par rapport à un autre.

### **Conclusion :**

L'activité de ce chapitre est de mettre en place une modélisation et une conception du système de e-maintenance, nous avons pu intégrer le modèle issu de l'approche Alaadin et la fonction de pilotage des systèmes industriels pour en extraire un modèle qui est la base de notre application.

Les résultats de ce chapitre ont servi de base pour le début de la partie réalisation et implémentation de l'application.

# Chapitre IV

---

Réalisation & Implémentation

### Introduction

Ce chapitre nous permet de montrer les résultats de notre application dans sa phase de test et d'essai, ce qui nous permettra d'envisager les améliorations possibles. Mais tout d'abord nous allons commencer par une brève présentation des outils de développement utilisés.

### 1. Les outils utilisés

#### 1.1 Langage utilisé :



Langage JAVA, C'est un langage de programmation orienté objet, développé par Sun Microsystems depuis racheté par Oracle Corporation. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels.

Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp) (**Arnold**)

#### 1.2 IDE Utilisé :



NetBeans IDE est un environnement de développement intégré open-source. NetBeans IDE prend en charge le développement de tous les types d'applications Java (Java SE (y compris JavaFX), Java ME, web, EJB et applications mobiles) hors de la boîte. Parmi les autres caractéristiques sont un système basé sur Ant projet, support Maven, refactorings, contrôle de version (soutien CVS, Subversion, Git, Mercurial et Clearcase). (**NetBeans, s.d.**)

### 1.3 Serveur Utilisé :



WampServer (**wampserver, s.d.**) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL

La version 2.4.9 utilisé dans ce projet intègre Apache 2.4.17, MySQL 5.7.9, PHP 5.6.16 et PHP 7, PhpMyAdmin 4.6.2, SQLBuddy 1.3.3, XDebug 2.2.3.

### 1.4 Connecteur JDBC (Java Database Connectivity) MySQL :

JDBC (Java Database Connectivity) est une interface de programmation créée par Sun Microsystems depuis racheté par Oracle Corporation-, pour les programmes utilisant la plateforme Java. Elle permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC. Normalement, il s'agit d'une base de données relationnelle, et des pilotes JDBC sont disponibles pour tous les systèmes connus de bases de données relationnelles (**technetwork java overview, s.d.**)

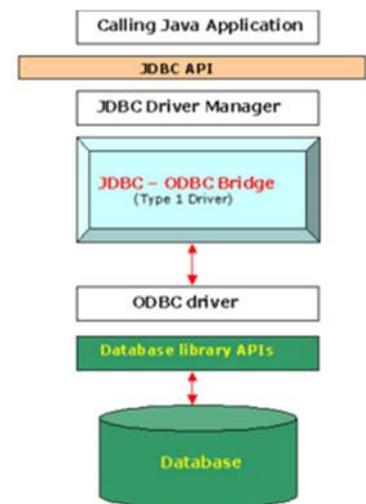


Figure 1: JDBC Connector

### 1.5 SGBD :

Pour implémenter notre base de données, on a utilisé l'outil PhpMyAdmin fourni avec le logiciel Wampserver. Le SGBD PhpMyAdmin, avec son interface graphique, nous a permis de créer et de gérer l'ensemble des tables de notre base de données (**Delisle, 2005**)

PhpMyAdmin, consiste en un ensemble de scripts PHP, permettant d'administrer des bases de données MySQL via un navigateur Web. Il a Plusieurs fonctions parmi elles, on cite :



- La création et la suppression de bases de données.
- La création, la modification et la suppression de tables.

## Chapitre IV : Réalisation & Implémentation

- L'édition, l'ajout et la suppression de champs.
- L'exécution de commandes SQL et de requêtes batch.
- La création d'index.
- Le chargement de fichiers textes dans des tables.
- La gestion des privilèges des utilisateurs JDBC.

### 1.6 Agent Sniffer

L'agent sniffer est utilisé lorsqu'un utilisateur décide d'épier un agent ou un groupe d'agents « écouter, débbugger, sauvegarder en fichier les conversations multi-agent ». Pour cela chaque message partant ou allant vers ce groupe est capté et affiché sur l'interface du sniffer. L'utilisateur peut voir et enregistrer tous les messages, pour éventuellement les analyser plus tard. (MECIBAH, 2012)

L'agent sniffer utilise une notation proche d'UML et L'agent peut être lancé du menu du RMA ou de la ligne de commande suivante :

`Java jade.Boot sniffer:jade.tools.sniffer.sniffer`

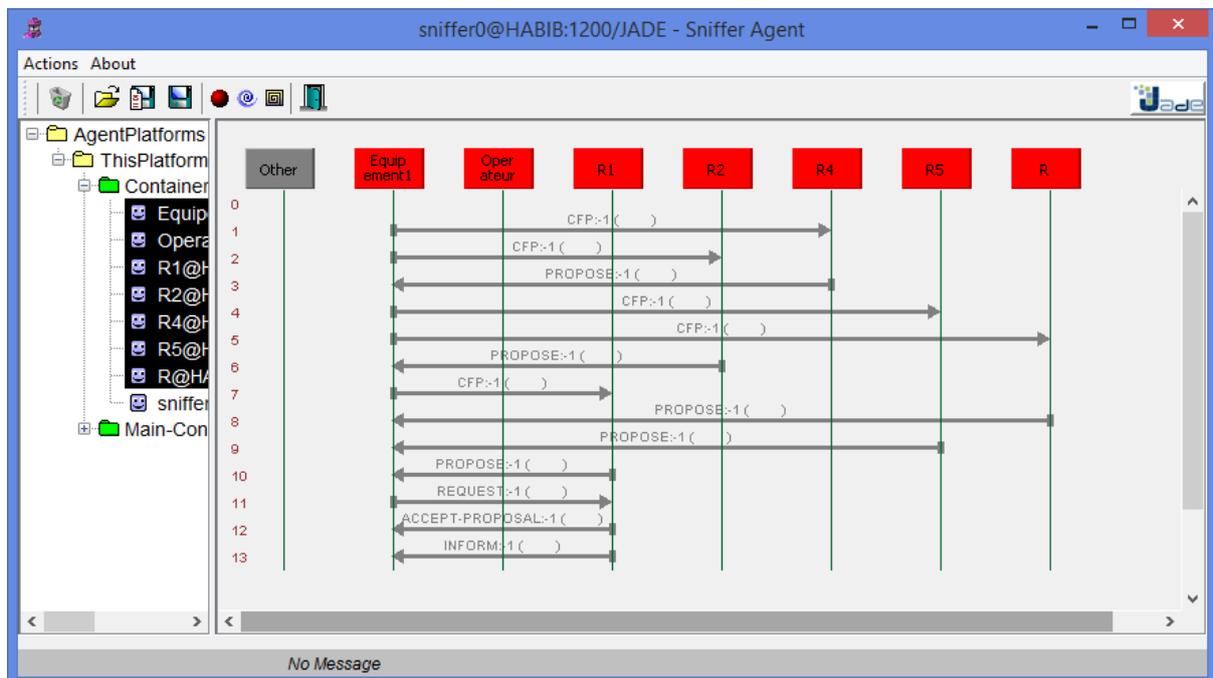


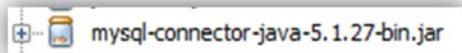
Figure 2: Agent Sniffer

### 1.7 Les bibliothèques Utilisé :

Avec l'utilisation de l'environnement Netbeans et pour le fonctionnement correct de notre application certain bibliothèque sont requis tels que JDBC, JADE, JFreeChart.

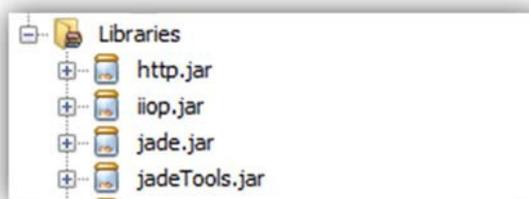
### a. JDBC

La bibliothèque JDBC contient une classe pour la connexion de la base de données.



### b. JADE

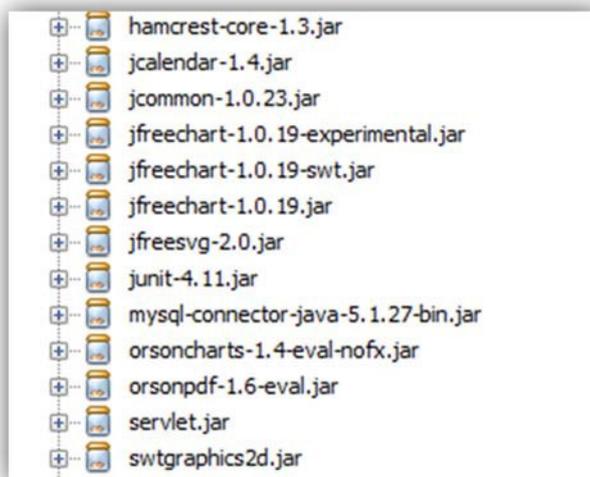
La bibliothèque JADE contient quatre classes (http, iiop, jade, jadeTools) pour la facilitation de l'interaction avec les agents.



*Figure 4: Jade Lib*

### c. JFreeChart

La bibliothèque JFreeChart "Java chart Library" contient 11 classes qui le rend facile pour d'afficher des graphiques de qualité professionnelle dans notre application.



### 2. Etapes de Conception du notre système

La conception de notre système comporte plusieurs étapes à savoir : Création des bases de données, présentation, la fonctionnalité.

#### 2.1 Création des bases de données

On a utilisé dans ce projet une base de données nommée "**maintenance**" qui contient sept tables : **Categorie**, **Panne**, **Reparateur**, **Piece**, **Piecepanne**, **Reparation** et **Equipement**. Pour cela on a opté pour un SGBD MySQL intégré à Wampserver.

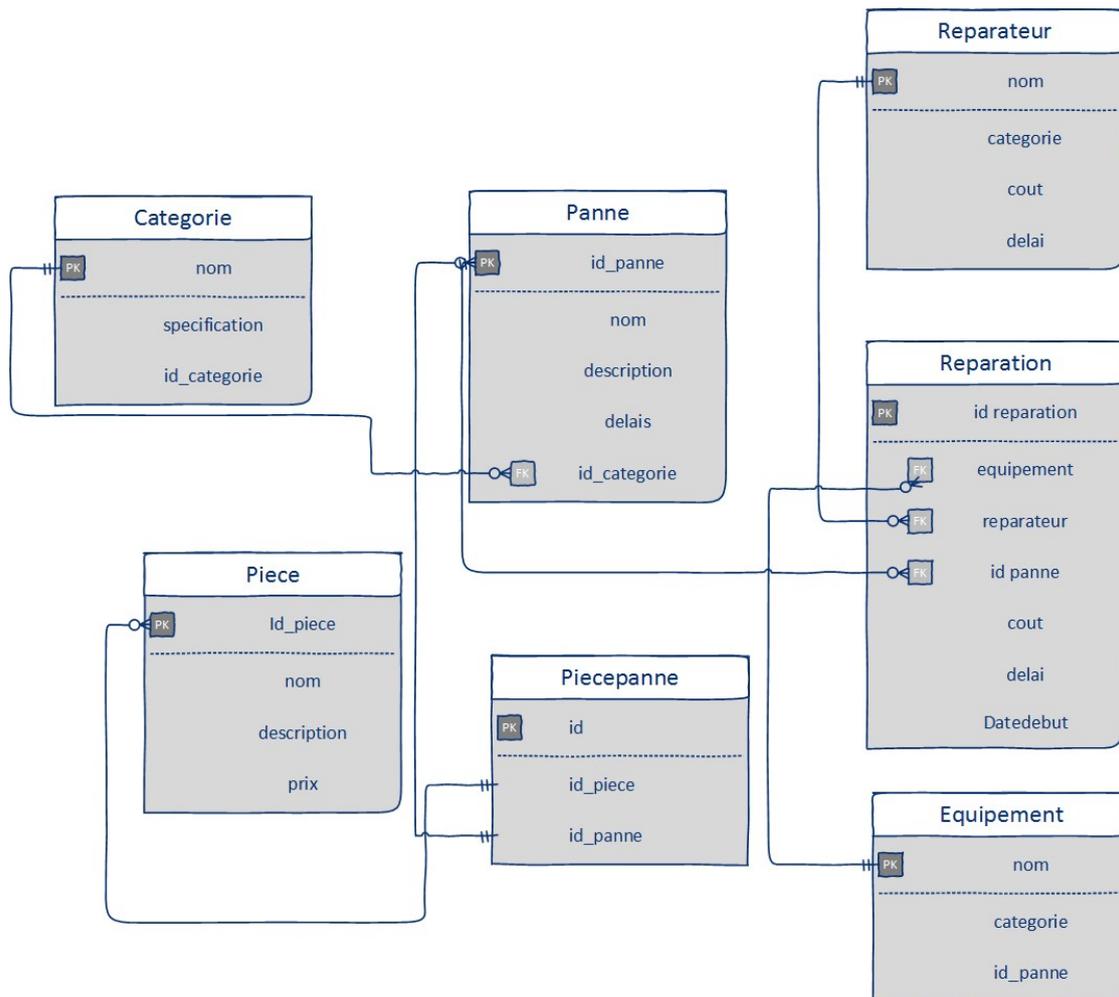


Figure 6: Base de données " Maintenance "

- **Table equipement :**

La table "Equipement" représente les agents équipements contenant le **nom** comme clé primaire, **catégorie** et dans notre cas chaque équipement contient une seule panne **id\_panne** .

- **Table reparateur :**

La table "Reparateur" représente l'agent ressource contenant le **nom** comme clé primaire, **catégorie** et **cout** horaire

- **Table panne :**

La table "Panne" représente les pannes des équipements contient les attribues suivant **id\_panne**, **nom**, **description**, **délais**, **id\_categorie**.

- **Table piece :**

La table "Piece" représente les pièces détachés contient les attribues suivant **id\_piece** comme clé primaire, **nom** de la pièce, **description** et **prix** de la pièce.

- **Table Categorie :**

La table "Categorie" représente les types de maintenance par exemple : une maintenance Mécanique, Electricité, Taulier. Cette table continente les attribue suivant **nom** comme clé primaire **specification** et **id\_categorie** comme clé étrangère.

- **Table piecepanne :**

La table "piecepanne" représente un lien entre les pannes et les pièces détachées pour la sélection des pièces déjà existant en la création de nouvelles pannes. Cette table contient les attribues suivant **id** comme clé primaire **id\_piece**, **id\_panne** comme des clés étrangers.

- **Table reparation :**

La table "reparation" représente les équipements en réparations de temps réel cette table contient les attribues suivants : **id\_reparation** comme clé primaire, **equipement**, **reparateur**, **id\_panne** comme des clés étrangers et **cout**, **delai**.

## 2.2 Présentation

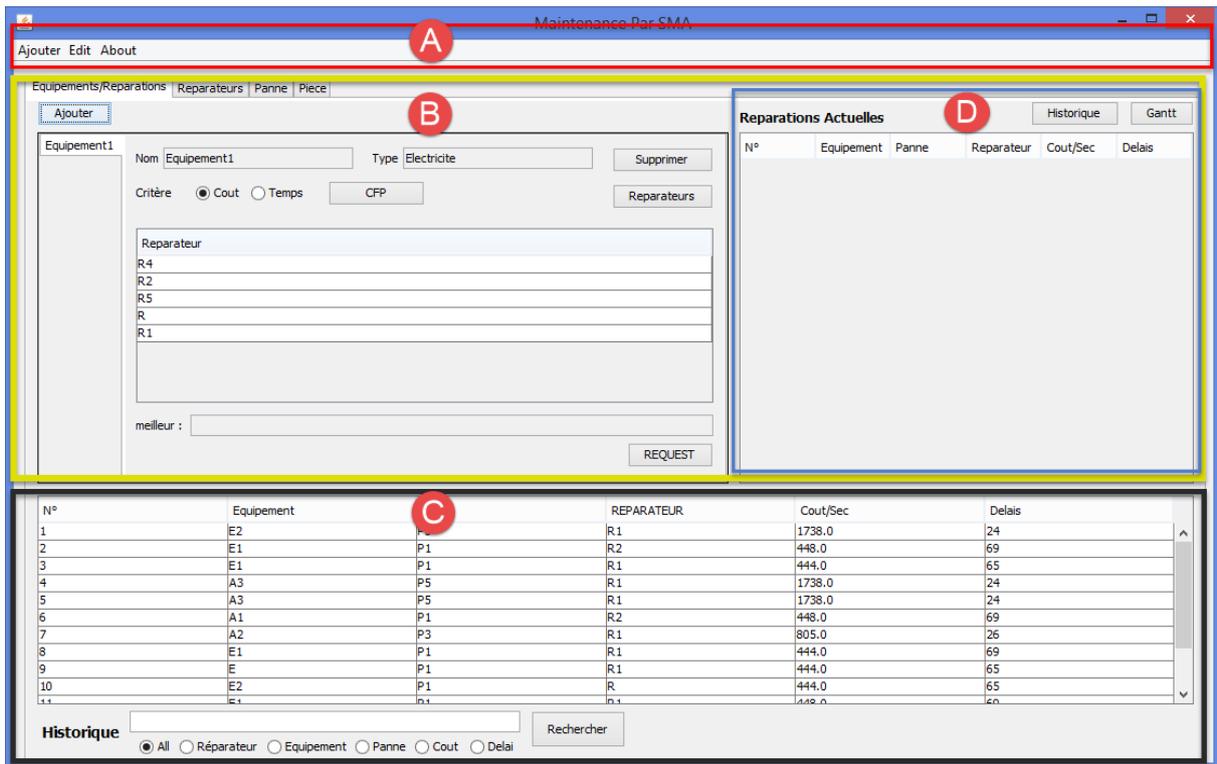


Figure 7: Interface principale

Figure 8 montre l'interface principale qui apparaît dès le lancement de l'application. Nous allons décomposer la fenêtre principale en trois parties :

- la partie (A) : une barre de menu qui contient des raccourcis vers les fonctionnalités de l'application.
- La partie (B) : ce sont les onglets ou les opérations principales se déroulent
- La partie (C) : présente un tableau contenant l'historique des opérations effectués.
- La partie (D) : ce volet contient un tableau des réparations actuelles

### La partie (A)

Contient seulement des raccourcis vers des options qu'on va illustrer en dessus et le volet à propos.

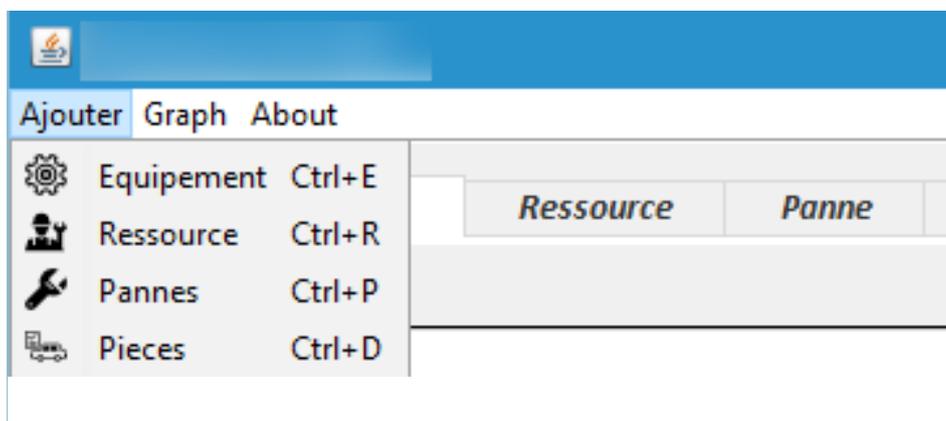


Figure 10 est un aperçu de l'agent sniffer sur notre système cette fenêtre apparaît avec le lancement de l'application.

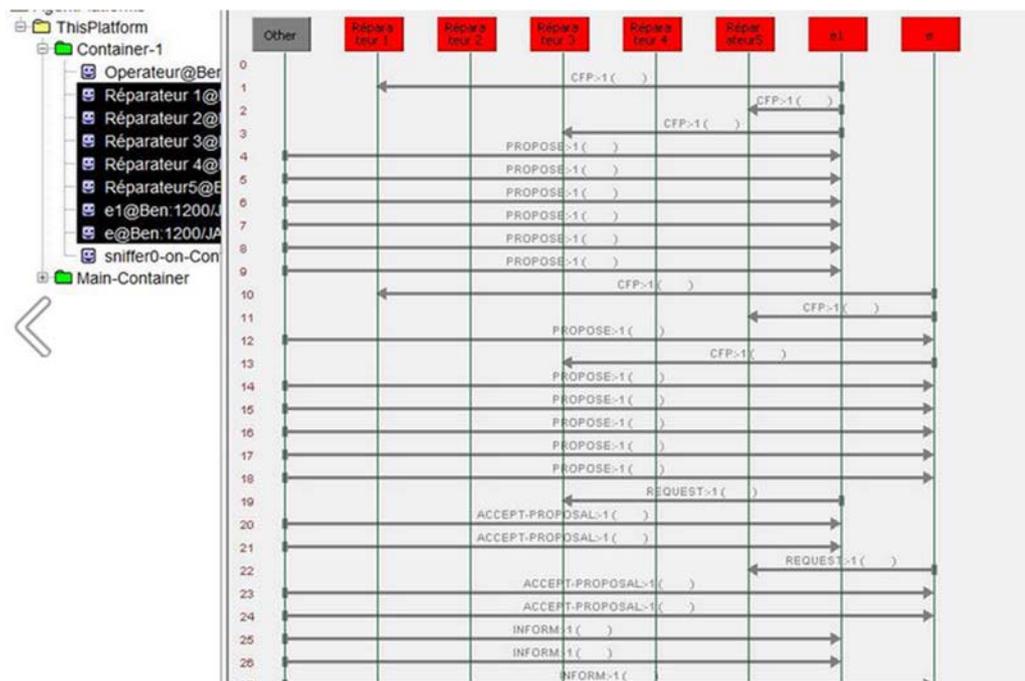


Figure 9: Communication & négociation

## 2.3 Fonctionnalité

### La partie (B)

#### L'onglet Réparation :

Dans ce qui suit nous allons détailler la partie B qui est la partie fonctionnelle du système. Le volet B comprend quatre volets ① gestion de la réparation ② gestion des Ressource ③ gestion des Pannes ④ gestion des Pièces

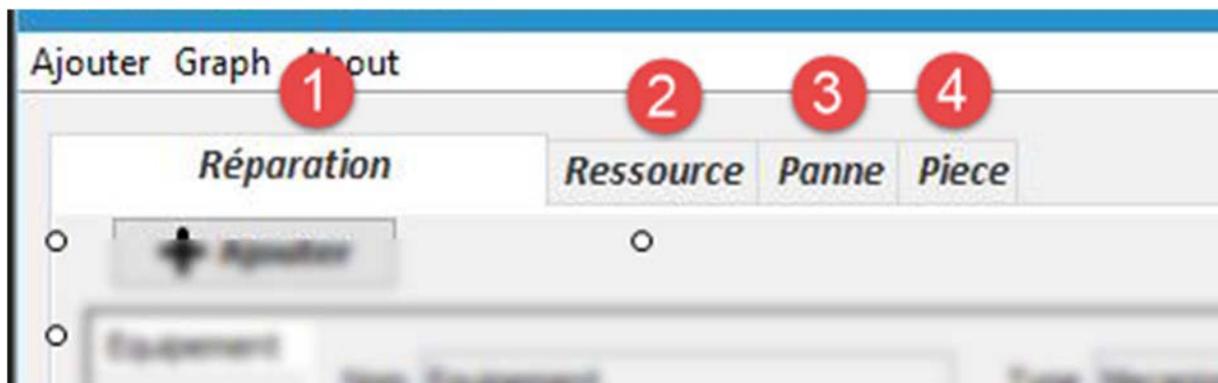


Figure 10: les onglets de l'application

## Chapitre IV : Réalisation & Implémentation

Nous allons illustrer l'onglet ① mène au panneau où se déroule les opérations de communication et de négociation. La figure 12.

En appuyant sur l'onglet ② nous recevons ⑥ ce bouton permet de lancer un nouveau équipement à maintenir réparation figure 12 ⑦ le nom de l'équipement concerné par la réparation ⑧ les critères choisis pour l'appel d'offre ⑨ bouton du lancement de l'appel d'offre ⑩ raccourci pour créer un nouveau réparateur ⑪ le contractant retenu par l'appel d'offre ⑫ confirmer le lancement de l'opération ⑬ liste des participants dans l'appel d'offre ⑭ les équipements en pannes à maintenir.



Figure 11: Equipement

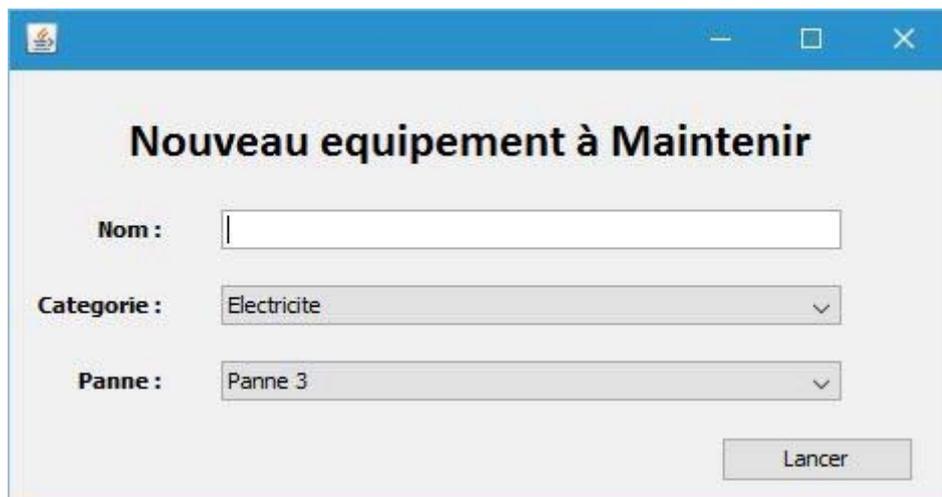


Figure 12: l'ajoute d'un nouvel équipement à maintenir

### L'onglet Ressource :

①⑥ permet l'ajout de nouveaux réparateurs figure 15 et ①⑦ la suppression ①⑧ actualisation du tableau ①⑨ la liste des réparateurs existants dans la base de données

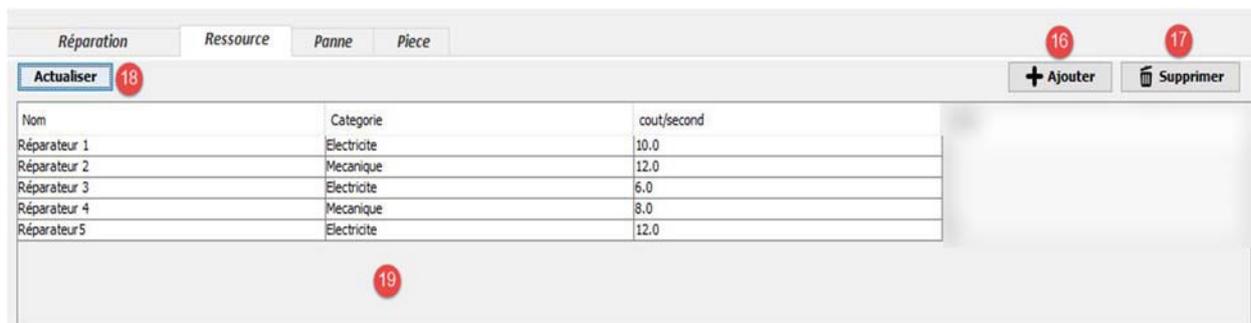


Figure 14: onglet des ressources

Nouvelle Ressource

Nom :

Categorie :

Cout :

delais :

Lancer

Figure 13: Ajouter nouvelle ressource

### L'onglet panne :

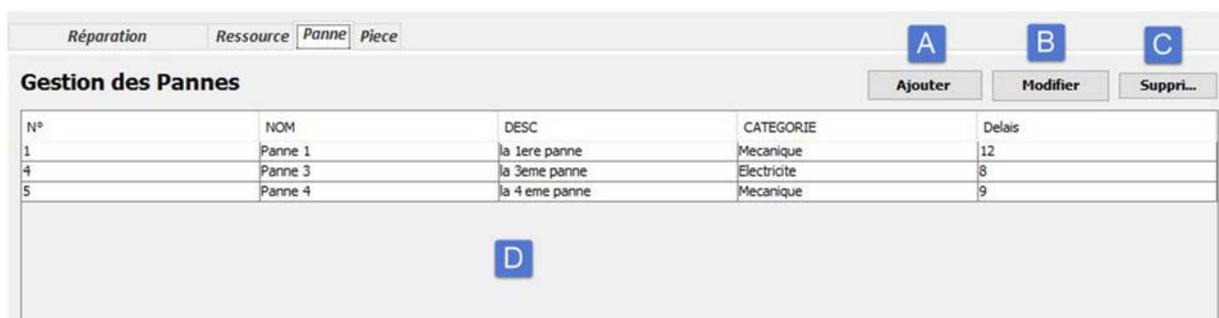


Figure 15: Gestion des pannes

## Chapitre IV : Réalisation & Implémentation

Figure 16 présente l'onglet gestion de panne ou on trouve : [D] tableau contenant les pannes existantes créé au préalable [A] bouton permet l'ajout de nouvelle panne et qui mène à un pop-up Figure 18 [B] modifier un panne existante par le biais d'une nouvelle fenêtre Figure 17 [C] supprimer une panne.

**Ajouter Une Panne**

ID : 4

Nom :

Categorie: Electricite

Delai :

Description :

Pieces :

+ Ajouter X Annuler

Figure 17: Ajouter une panne

**Modifier Une Panne**

ID : 4

Nom : Panne 2

Delai : 11

Categorie: Electricite

Description : 1a 2eme panne

Pieces :

| N° | nom_pi... | descript... | prix |
|----|-----------|-------------|------|
| 8  | p1        | p1p1p1p...  | 11.0 |

Modifier X Annuler

Figure 16: modifier une panne

### L'onglet pièce :

Réparation Ressource Panne **Pièce**

**Gestion des Pièces** A B C

Ajouter Modifier Supprimer

| N° | NOM | DESC          | PRIX |
|----|-----|---------------|------|
| 5  | p1  | la 1ere Piece | 11.0 |
| 6  | p2  | La 2eme piece | 15.0 |
| 7  | p3  | la 3eme piece | 3.0  |

D

Figure 18: Gestion des pièces

Figure 19 présente l'onglet gestion de pièces ou on trouve : [D] tableau contenant les pièces existantes dans la base de données [A] bouton permet l'ajout de nouvelle panne et qui mène à un pop-up Figure 20 [B] modifier une pièce existante par le biais d'une nouvelle fenêtre Figure 21 [C] supprimer une pièce.



### Modifier Une Piece

ID : 8

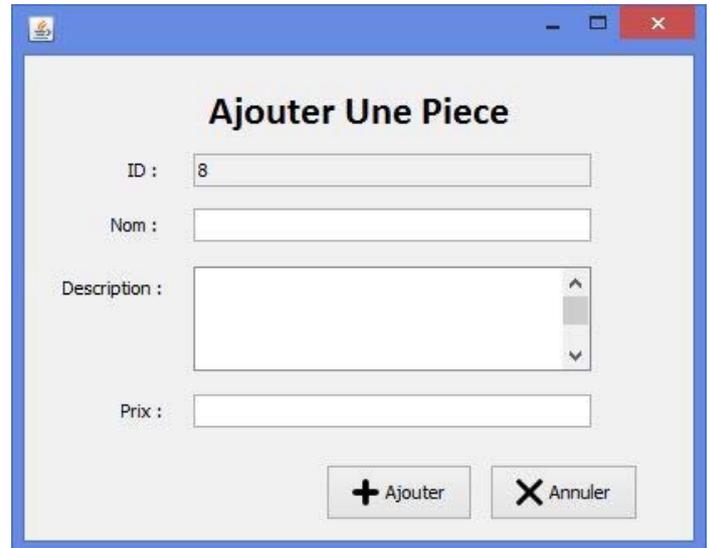
Nom : piece1

Description : piece 1

Prix : 14.0

Modifier Annuler

Figure 20: Modifier une Pièces



### Ajouter Une Piece

ID : 8

Nom :

Description :

Prix :

+ Ajouter X Annuler

Figure 19: Ajouter une pièce

### La partie (C)



## Historique

All  Réparateur  Equipement  Panne  Cout  Delai

Rechercher GANTT

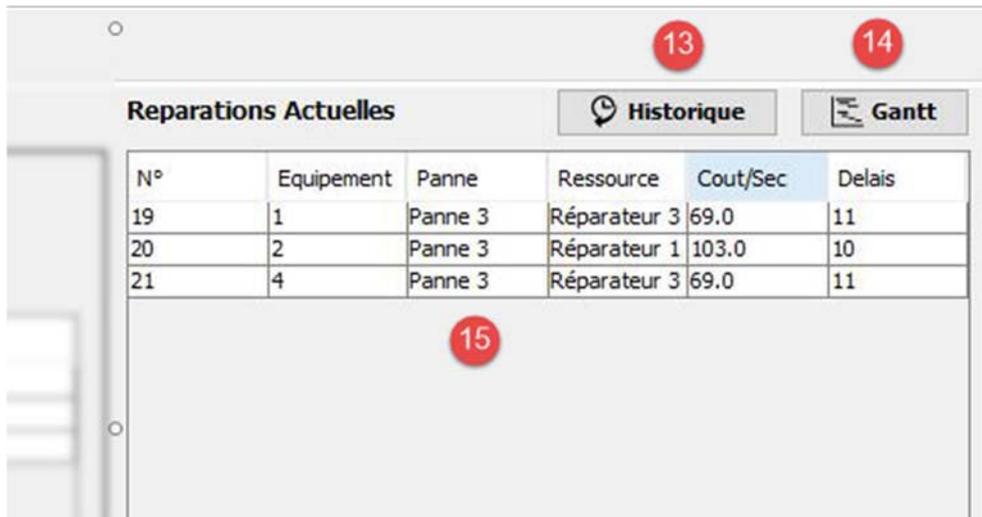
| N° | Equipement   | Panne   | Ressource    | Cout/Sec | Delais |
|----|--------------|---------|--------------|----------|--------|
| 1  | Equipement1  | Panne 3 | Réparateur 3 | 69.0     | 11     |
| 2  | Equipement2  | Panne 3 | Réparateur 1 | 103.0    | 10     |
| 3  | Equipement3  | Panne 1 | Réparateur 4 | 141.0    | 14     |
| 4  | Equipement4  | Panne 1 | Réparateur 2 | 233.0    | 17     |
| 5  | Equipement5  | Panne 1 | Réparateur 4 | 141.0    | 14     |
| 6  | Equipement6  | Panne 3 | Réparateur 1 | 103.0    | 10     |
| 7  | Equipement7  | Panne 1 | Réparateur 2 | 233.0    | 17     |
| 8  | Equipement8  | Panne 3 | Réparateur 3 | 69.0     | 11     |
| 9  | Equipement9  | Panne 1 | Réparateur 4 | 141.0    | 14     |
| 10 | Equipement10 | Panne 1 | Réparateur 2 | 233.0    | 17     |
| 11 | Equipement11 | Panne 3 | Réparateur 1 | 103.0    | 10     |
| 12 | Equipement12 | Panne 3 | Réparateur 3 | 69.0     | 11     |

Figure 21: Historique

Figure 22 montre la fenêtre historique, ou liste des opérations effectués apparaît dans un tableau ④ nous avons aussi mis une barre de recherche, ① des radio box pour sélectionner le filtre de la recherche. Le bouton ② lance la recherche et le ③ affiche la fenêtre du diagramme de Gantt

### La partie (D)

La figure 23 présente La partie (D) concernant le suivi des opérations après le lancement de l'appel d'offre.



The screenshot shows a window titled 'Reparations Actuelles'. At the top right, there are two red circular callouts: 13 pointing to the 'Historique' button and 14 pointing to the 'Gantt' button. Below these buttons is a table with the following data:

| N° | Equipement | Panne   | Ressource    | Cout/Sec | Delais |
|----|------------|---------|--------------|----------|--------|
| 19 | 1          | Panne 3 | Réparateur 3 | 69.0     | 11     |
| 20 | 2          | Panne 3 | Réparateur 1 | 103.0    | 10     |
| 21 | 4          | Panne 3 | Réparateur 3 | 69.0     | 11     |

Below the table, there is a large empty area with a red circular callout 15 in the center.

Figure 22: Réparations actuels

Le bouton 13 mène à l'historique des opérations c'est un raccourci historique [partie (C)] 14 diagramme de Gantt présenté dans la figure 23 15 la liste des réparations en cours de réalisation

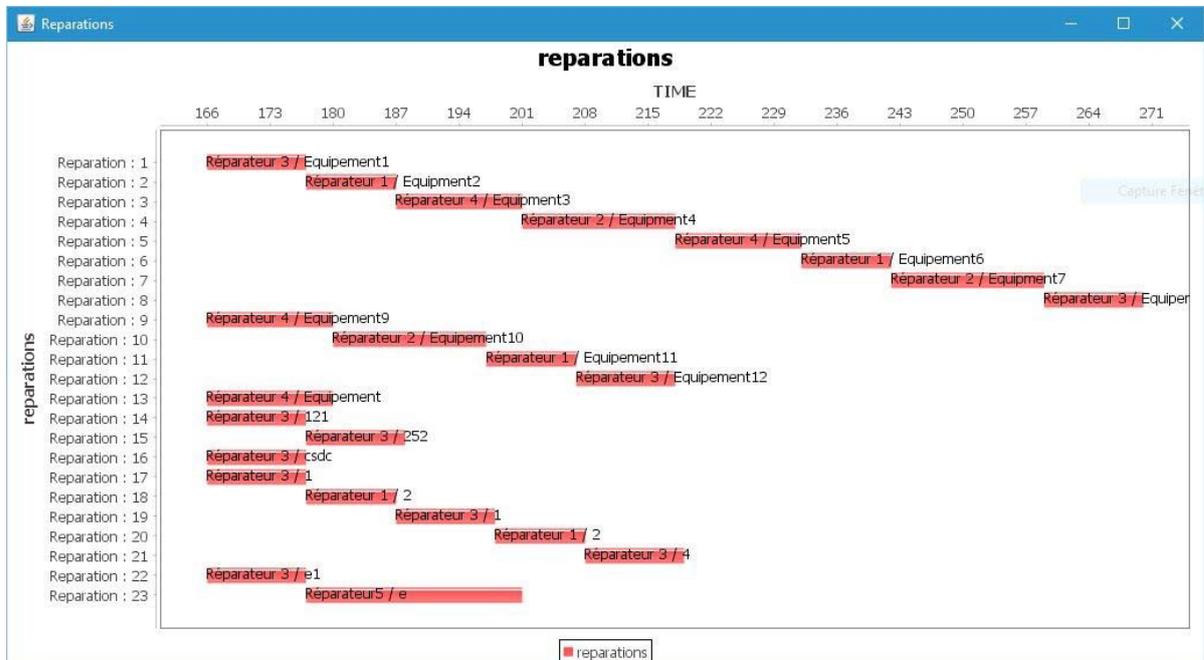


Figure 23: Diagramme de Gantt

## 3. Modèle d'utilisation du système conçu :

La figure 24 ci-dessus montre un modèle d'utilisation du prototype dans un environnement industrielle.

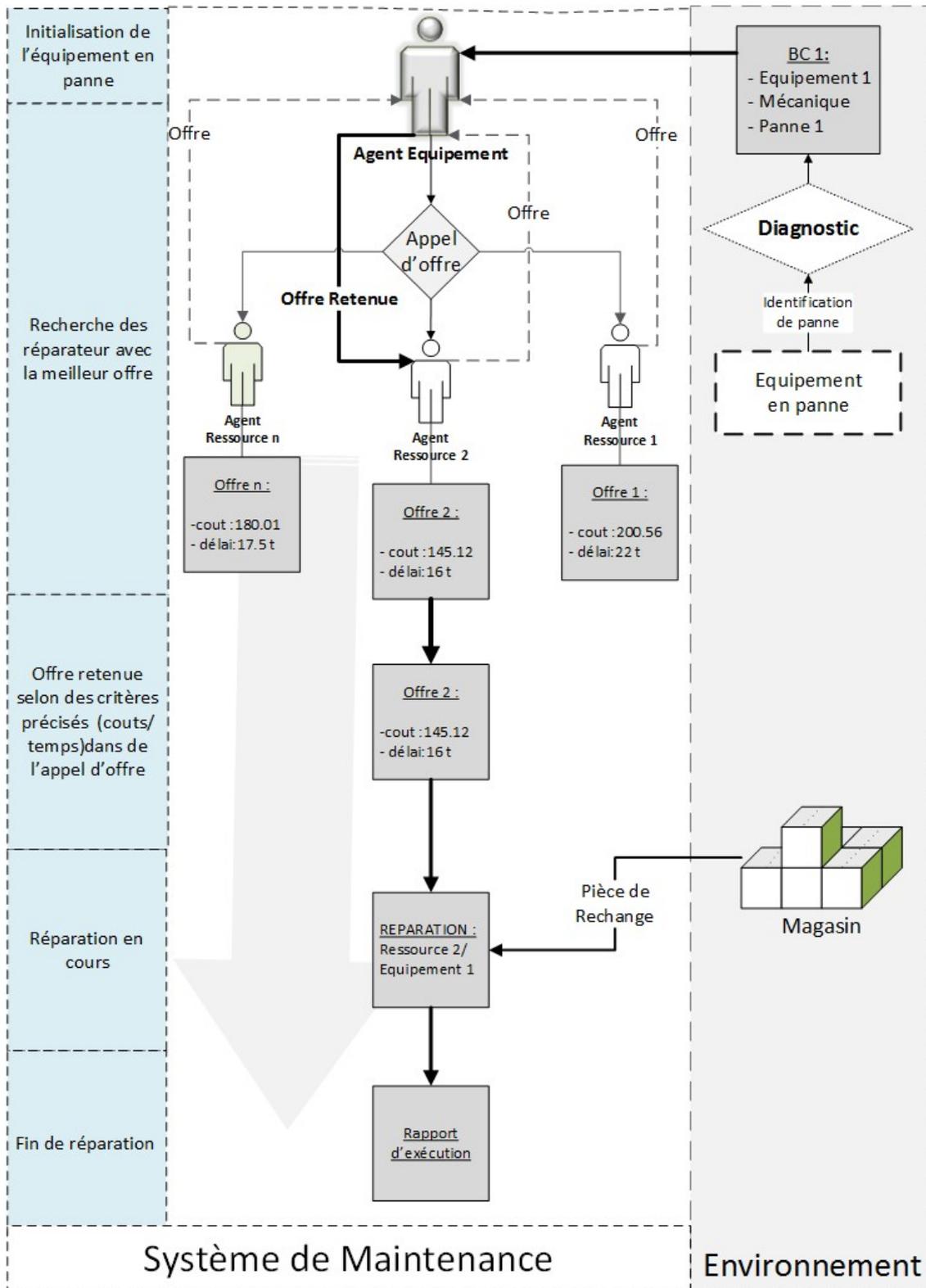


Figure 24 : Modèle d'utilisation du prototype

### Conclusion

A la fin de ce chapitre, nous avons atteint la fin de ce travail. Nous avons vu les différents aspects pratiques du système de maintenance réalisé ; Nous avons aussi expliqué la manière dont ce prototype peut être utilisé par des illustrations et des figures de l'application mise en œuvre.

## Conclusion générale

Dans ce projet nous avons entamé le domaine de la maintenance industrielle en citant ces aspects et ces caractéristiques ; On a aussi abordé l'évolution des systèmes de maintenance et leurs relations avec l'évolution technologique en informatique.

Le but fondamental est d'arriver à concevoir et à modéliser un système Multi agent conforme au concept des systèmes de l'e-maintenance.

Tout d'abord nous avons adopté l'approche du modèle Alaadin pour ces atouts, nous avons aussi proposé un modèle de pilotage de maintenance basé sur le pilotage des systèmes industriels qui comprend quatre modules : planifier, exécuter, réagir et suivre ; Nous avons développés les deux modules « planifier » et « réagir » en les transformant aux modules « négociation » et « communication ».

On a adopté deux protocoles pour gérer les interactions des agents intervenants dans ce système, le protocole « contract Net » et le protocole « Request », le premier est utilisé pour administrer les négociations entre agents et le deuxième pour gérer la communication ; après la phase de l'implémentation nous avons eu comme résultat un prototype de système de maintenance gérant les interactions des intervenants.

Durant la réalisation de ce projet on s'est heurté à une complication dans la compréhension du modèle Alaadin et la fonction de pilotage des systèmes industriels, plus des difficultés dans la gestion des interfaces graphique. Nous avons décortiqué les deux modules à savoir les modules « planifier » et « réagir » et on a conclu que le traitement de la fonction de pilotage de système de maintenance peut rendre le système de la e-maintenance fonctionnel dans un site de maintenance.

Nous prenons pour des perspectives ouvertes le développement des autres deux modules de la fonction du pilotage des systèmes industriels à voir « exécuter » et « suivre » ; et rendre ce prototype de la e-maintenance fonctionnel dans un site de maintenance.

## Références

- A.Chadli. (n.d.). BDI + Multi-Niveaux 2.
- AFNOR. (n.d.). *Les différents types DE MAINTENANCE*. Retrieved from <http://www.boutique.afnor.org>: <http://www.boutique.afnor.org/extraits/FA092125.pdf>
- Arnold. (n.d.). *Le langage Java. International Thomson Publishing Présentation guidée claire du langage.*
- B, A. (1991). *conception d'une architecture reactive distribué et hiérarchique pour le pilotage des systemes de production. thèse de Doctorat en science.*
- B.ESPINASSE. (2012). *Coordination et Négociation dans les SMA. LSIS UMR CNRS.*
- Brown, H. P. (1990). *The Monitoring in production activity Control.*
- CMMS. (n.d.). *micromain*. Retrieved from <http://www.micromain.com/the-cmms-software-hunt/>
- Delisle, M. ( 2005). *Gestion de bases de données avec phpMyAdmin. Campu Press.*
- Ferber. (1995). *Les Systèmes Multi-Agents, vers une intelligence collective.*
- Florea, A. M. (n.d.). *Agents et Systèmes Multi-agents*. Retrieved from turing: <http://turing.cs.pub.ro/auf2/>
- Franklin, G. &. (1996.). *Is it an Agent, or just a Program?:A Taxonomy for Autonomous Agents.*
- GMOA. (n.d.). Retrieved from wikipedia: [https://fr.wikipedia.org/wiki/Gestion\\_de\\_maintenance\\_assistee\\_par\\_ordinateur](https://fr.wikipedia.org/wiki/Gestion_de_maintenance_assistee_par_ordinateur)
- Gutknecht Olivier. (2001). *Actes des septièmes Journées Francophones d'Intelligence Artificielle Distribuée et. Vers une méthodologie organisationnelle de conception de systèmes.* Montpellier.
- Gutknecht, F. (1998). *la conception et l'exécution de systèmes multi-agents. ICMAS.*
- Gutknecht, J. F. (1998). *A meta-model for the analysis and design of. IEEE CS Press, pages 128–135.*
- H.SIABDELHADI. (n.d.). *Prototypage et validation du modèle SMA pour le pilotage des système de production.*
- Herrman, J. (2010). *rapport de stage.*
- <http://www.selectron.ch/en/solutions/remote-maintenance.php>. (n.d.). Retrieved from <http://www.selectron.ch>: <http://www.selectron.ch/en/solutions/remote-maintenance.php>
- Ivana Rasovska, B. C.-M. (2007). – 5-8 juin 2007 – . *7e Congrès international de génie industriel, (p. 9).* Trois-Rivières, Québec (CANADA).
- jones, A. T. (1990). *multi Algent multi layer architecture for intelligent control. international journal of computer Integrated Manufacturing.*
- J-P, M. (1998). *Analysis and design of multi-agent systems. First SIG meeting.*