République Algérienne Démocratique et Populaire

وزارة التعليم العالـي و البحـث العلمـي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf

***F**aculté de **M**athématiques et **I**nformatique*

***D**épartement d'**I**nformatique*

MEMOIRE EN VUE DE L'OBTENTION DU DIPLOME DE MAGISTERE

Spécialité : *Informatique*
Option : *MOEPS*

Présenté par : ***Mr SALAH Djilali***

Thème

LA CONTRIBUTION DES SYSTEMES MULTI-AGENTS DANS LA TECHNOLOGIE DU DATA MINING DISTRIBUE

Soutenu le 20/06/2016

Le jury est composé de :

➢ **Pr**	**BENYETTOU A.E.K.**	**Président**	**USTO-MB, ORAN**
➢ **Dr**	**RAHAL S.A.**	**Rapporteur**	**USTO-MB, ORAN**
➢ **Pr**	**BENDELLA F.**	**Examinatrice**	**USTO-MB, ORAN**
➢ **Dr**	**BELKADI K.**	**Examinateur**	**USTO-MB, ORAN**
➢ **Dr**	**KHIAT S.**	**Invité**	**ENPO, ORAN**

Année Universitaire 2015 / 2016

Remerciement

Je remercie tout d'abord ALLAH pour m'avoir donné la force et la volonté car sans lui rien n'aurait pu être, je remercie après mes chers parents, mes frères et sœurs qui sont mon capital et ma source principale d'inspiration.

Je remercie ensuite mon encadreur et mes valeureux enseignants qui m'ont aidé et soutenu quelque soit les circonstances, citons Mr RAHAL, Mr Khiat, Mr Belkadi, Mr Benyettou et d'autres que je ne peux pas tous les citer.

Je dis Mercie également à tous mes collègues de la Wilaya d'Oran Mr Abdelghani Filali (SG), Mr Lasfar Brahim et Mr Khelil ainsi que mon ami Mr Mestar Amine et tous les autres.

Je n'oublierais pas aussi de remercier mes amis réels et virtuels qui n'ont jamais oublié la vraie valeur de l'amitié et qui ne m'ont jamais laissé tomber.

Enfin, je tiens à remercier l'ensemble des personnes qui m'ont soutenu et aidé de prés ou de loin et surtout ceux que je n'ai pas cité

Dédicace

Je dédie ce travail en premier lieu à ma famille proche et lointaine, à mon encadreur, à mes amis et mes collègues.

Je dédie ce travail également aux futur titulaires de Magistère qui devront à leurs tours préparer une thèse de Recherche et qui partageront les mêmes idées que moi.

Je dédie ce travail enfin aux membres de jury qui vont me supporter pendant une heure pour m'aider à accomplir mon cycle de Post-Graduation et Obtenir enfin Mon Magistère en informatique

La Table des Figures

La Liste des Tableaux

Table des Matières :

Introduction Générale

Introduction Générale

L'utilisation fréquente de la technologie des bases de données dans divers domaines comme les réseaux de communication informatique centralisés et distribués, homogènes et des systèmes utilisant de multiples bases de données, cela a conduit au développement de systèmes multi bases de données. Or, en ce qui concerne la prise de décision, de grandes organisations ont vu la nécessité de faire appel aux techniques avancées pour fouiller ces bases de données centralisées ou distribuées dans leurs profondeurs et dans leurs branches.

Pendant que les processus de data mining se déroulent normalement et aboutissent à des motifs fréquents permettant de modéliser des résultats compréhensibles, les processus de fouilles qui s'appliquent sur des environnements distribués font naissance à de nouvelles problématiques. Par conséquent, de nouvelles stratégies de fouille multi-bases de données adaptables aux systèmes distribués ont été mises en place pour aider à explorer de tel supports de données complexes.

Ce travail aborde certaines questions concernant l'utilisation de techniques de fouille de données dans un environnement distribué. Les algorithmes de recherche de règles d'association distribuées et les améliorations possibles apportées par l'intégration de la technologie des agents nous ont poussé à voir d'autres solutions, ce qui consiste à concevoir un système de fouille de données basé sur une architecture bien connue et essayer de l'optimiser à l'aide de la technologie des agents. Ainsi, nous pouvons faire une comparaison entre les performances de l'ancien système adopté et celui du système obtenu après l'amélioration.

Le mémoire est constitué de quatre chapitres décrits comme suit :

Le premier chapitre :

Ce chapitre donne un rappel sur l'aspect général et historique du data mining et celui des systèmes multi agents tout comme il donne aussi une présentation sur quelques travaux récents basés sur le data mining combiné à la technologie des agents dans le cadre des données distribuées.

Le Deuxième chapitre :

Durant cette étape on va étudier les différents algorithmes distribués utilisés dans le data mining et voir leurs avantages et inconvénients. On va aussi décrire l'algorithme distribué qui va être adopté pour notre travail.

Le troisième chapitre :

Ici, on donnera une explication plus détaillée sur le fonctionnement de l'algorithme distribué qu'on a choisi pour notre étude. Ensuite, on va présenter une architecture qui servira de base pour notre système. Après cela, il faut faire une étude schématique et conceptuelle pour montrer ses performances théoriques et ses fonctionnalités.

Le Quatrième chapitre :

Le dernier chapitre consiste à décrire et expliquer les démarches de l'implémentation du système et sa mise en œuvre technique tout en mettant en valeur les performances et les améliorations apportées lors de sa personnalisation.

Etat de l'Art sur les Agents

Dans le Data Mining Distribué

-Introduction

-Rappel sur le du Data Mining

-Le Data Mining distribué

-Rappel sur les Systèmes Multi Agents

-Fusion entre Data Mining et SMA

-Problèmes Liés au Data Mining basé Agents

-Quelques Systèmes existants du Data Mining basé Agents

-Etude Comparative des systèmes DMBA

-Discussion et Synthèse

-Conclusion

Chapitre 01 : Etat de l'Art sur les Agents dans DataMining Distribué

I.1-Introduction :

Dans le domaine de l'informatique, les informations et les données manipulées par la population ont pris des aspects divers notamment le texte, le son, les vidéos, les images, les bases de données et d'autres. Le volume de ces dernières s'accroît avec le nombre de personnes physiques et morales, le nombre et le volume des dépôts de données augmentent aussi énormément.

Les techniques de recherches conventionnelles aidant les utilisateurs à consulter leurs sources de données ne sont plus assez efficaces avec ce volume d'informations important. Donc, Les chercheurs ont pensé à créer des techniques complexes permettant de minimiser les résultats de recherches qui ne laissent apparaitre que des ensembles de données de plus en plus significatifs aux besoins de l'utilisateur (connaissances).

I.2-Rappels sur le Data Mining :

a-Historique :

Il est très connu que depuis les premiers âges de la micro-informatique que les utilisateurs s'adaptent aux augmentations de volumes des données qu'ils accumulent pendant l'exercice de leurs différentes activités, ils ont toujours cherché à régler deux grandes problématiques :

1- Ayant rassemblé des volumes énormes de données voisinant les Tera octets, on a toujours cherché une manière plus facile de retrouver des informations précises parmi d'autres en exécutant des requêtes. Les systèmes de bases de données et les DatawareHouses (1960-1996) ont répondu à ce problème. [01]

2- La deuxième question consistait à trouver avec une manière efficace des informations significatives et pertinentes à partir d'un ensemble de données colossal en laissant le libre arbitre à la machine d'extraire des connaissances utiles. Le data mining est apparu en 1990 pour répondre à cette problématique.[02]

b-Définition :

L'exploration de données, connue aussi sous l'expression de fouille de données, ou extraction de connaissances à partir de données, « ECD » en français, « KDD » en anglais, a pour objet l'extraction d'un savoir ou d'une connaissance à partir de grandes quantités de données par des méthodes automatiques ou semi-automatiques. [02]

c-Les domaines d'application DataMining :

- Analyse de données et aide à la décision [01W]
  - Analyse de marché
  - Marketing ciblé, gestion des relations client, analyse des achats des clients, ventes croisées, segmentation du marché
  - Analyse de risque
  - Détection de fraudes
  - Prévision des Marchés
- Autres Applications [01W]
  - Text mining : news groups, emails, documents Web.
  - Optimisation des requêtes, Machine Learning (Auto-Apprentissage)…etc.

d-Les étapes d'un processus de Data Mining :

La création d'un modèle d'exploration de données fait partie d'un processus plus vaste qui va d'un ensemble de requêtes de consultation des données, la création d'un modèle afin d'y répondre et le déploiement du modèle obtenu dans un environnement de travail. Ce processus peut être décrit à l'aide des six étapes de base suivantes [01W , 04]:

1. Définition du problème **:** cette étape consiste à définir clairement le problème et à envisager les moyens d'utilisation des données pour apporter une solution au problème.
2. Préparation des données **:** l'opération consiste à consolider et à nettoyer les données identifiées à l'étape « Définition du problème ».
3. Exploration des données **:** cela consiste à explorer les données préparées en se servant des algorithmes décrits.
4. Création des modèles **:** on va générer le ou les modèles d'exploration de données. Vous allez utiliser les connaissances acquises à l'étape Exploration des données pour définir et créer les modèles.
5. Exploration et validation des modèles **:** dans cette phase on va explorer les modèles d'exploration de données créés et à tester leur efficacité.
6. Déploiement et mise à jour des modèles **:** consiste à déployer les modèles les plus efficaces dans un environnement de production.

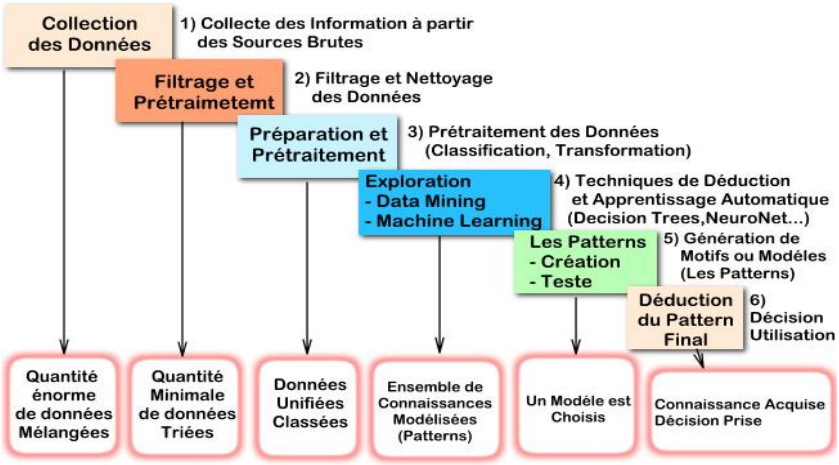


Figure1 : Les étapes d'un processus de fouille de données

e-Les Types d'apprentissage dans un processus de Fouille :

On a deux types de techniques d'apprentissage [01W, 01M, 05, 06] :

Apprentissage Supervisé : On a un ensemble d'objets A et un ensemble de classes C préalablement étiquetées, la méthode consiste à calculer la ressemblance des objets et les correspondre chacun à une classe. On obtient des classes de modèles où chaque nouvel objet doit être joint à une classe. Exemple : les arbres de décision.

Apprentissage Non Supervisé : On a un ensemble d'objet ou de données non étiquetées au préalable et ne possèdent aucune liaison entre elles, la technique consiste à appliquer des algorithmes spécifiques afin de trouver des relations et les associer à ces données.

I.3- Le Data Mining Distribué :

I.3.1-Définition :

Le datamining distribué (DDM : Distributed DataMining) se situe à la conjonction de deux évolutions majeures : d'une part, l'explosion de masses de données importantes et souvent réparties où il faut savoir extraire une connaissance utile, d'autre part l'utilisation de nouvelles heuristiques diminuant la complexité des traitements et plus aptes à une exécution parallèle, qu'en termes de distribution des traitements, des communications et des mémoires, dans un contexte non centralisé et hétérogène tout en minimisant l'interaction entre les différents sites [02M,07].

I.3.2-Principe :

Dans ce contexte, le modèle distribué propose une réponse aux problématiques imposées par le datamining centralisé :

- Avec un volume de données important et toujours croissant, il faut un support de stockage centralisé, exigeant en ressources et très couteux.
- Le transfert d'un ensemble de données depuis un site central vers d'autres (et réciproque) exige souvent des ressources importantes (bande passante, temps d'échange, mémoire I/O).
- Le calcul de données sur un site central est souvent exigeant en temps et en ressources par rapport à un calcul distribué.
- Il est dangereux de garder des données privées ou sensibles sur un seul site central exposé aux risques d'effacement, divulgation, modification par d'autres sites[02M,07].

I.3.3-Domaine d'Application :

L'Approche distribuée du DataMining partage le même domaine d'utilisation que le datamining classique, mais il est utilisé surtout dans les cas où on applique des fouilles dans un réseau, là où les postes sont limités du côté puissance et vitesse de traitement car il offre le parallélisme aux processus de datamining qui pourrait saturer les ressources d'une machine avec faible configuration lorsqu'elle l'exécute de manière individuelle [02M].

On note les domaines les plus marqués par le datamining distribué : Consultation Web, Finance, Médecine, Gestion Ressources Humaines, Laboratoires Pharmaceutique et Biotechnologiques…etc.

I.3.4-Les Règles d'association :

A)-Définition :

L'application de cette méthode est souvent ce qu'on appelle « l'analyse du panier de la ménagère », qui consiste à rechercher des associations entres des produits sur les tickets de la caisse, analyser les préférences des clients, marquer les produits qui ont une correspondance avec d'autres [02M, 07].
La recherche des règles d'association est une technique de datamining parmi les plus utilisées dans le marketing, on note qu'un système peut générer deux types de règles d'association :

-Des règles d'association classique sous la forme : si action1 ou condition1 alors action2
-Des règles d'association situées dans le temps qui sont sous la forme : si action1 ou condition1 à l'instant t1 alors action2 à l'instant t2.

Dans un support tel que les bases de données, la recherche des règles d'association est une manière évolutive et exploitable pour rassembler les liaisons entre différents items.

B)-L'extraction des Règles d'association :

C'est un des algorithmes appliqués lors de l'exécution du processus de fouille de données en mode non supervisé, on a comme résultat un ensemble d'ItemsSet. Ces derniers sont des ensembles d'éléments (Items) souvent corrélés positivement [02M, 07].

Dans certain cas, la recherche des règles d'association peut être aussi utilisée de manière supervisée. le plus important c'est que l'algorithme de base de l'extraction des règles d'association est l'Algorithme « A PRIORI » qu'on va décrire dans le prochain paragraphe [02M, 07].

C)-Algorithme A priori :

L'algorithme A-priori est un algorithme d'exploration de données conçu en 1994, par *Rakesh Agrawal* et *Ramakrishnan Sikrant*, dans le domaine de l'extraction des **règles d'association**. Il sert à reconnaître des propriétés qui reviennent fréquemment dans un ensemble de données et d'en déduire une catégorisation [07].

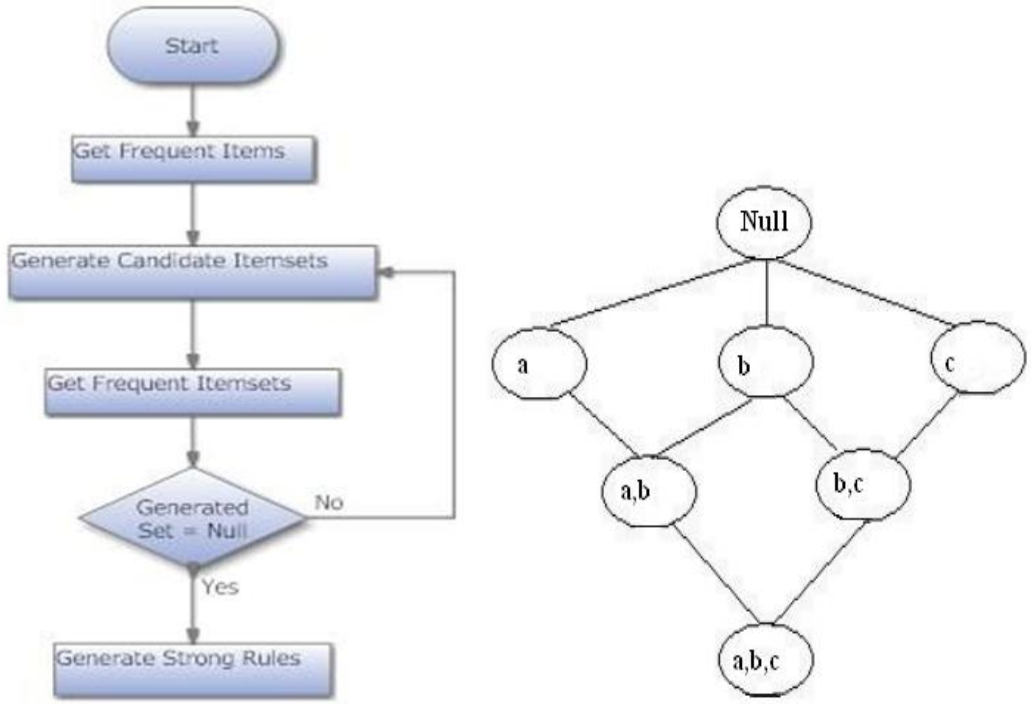


Figure2 : L'Algo. APRIORI (Gauche), un graphe d'ensemble d'Items (Droite)

La difficulté consiste notamment à trouver des règles qui soient significatives et non seulement le résultat du hasard. Exemples d'utilisation [08, 09] :

- Système de contrôle des achats sur une série de magasins
- Système de vérification d'identification
- Réseaux de Télécommunication : Détection d'anomalies, détection d'abus et de fraudes,
- Recherche Médicale : détection et prédiction de maladies liés aux relations consanguines ou autres causes.
- Domaine de l'industrie et automatisme : Surveillance des processus industriels et chaines stochastiques, prédiction de pannes.
- Analyse des données spatiales.

d)-Les étapes d'un processus de Recherche de Règles d'association :

Un processus de recherche de règles d'association vient souvent comme suit [08, 06W] :

- **Sélection et préparation des données :** Durant cette étape, on va rassembler les données sous forme d'attributs et d'objets des bases de données prévues à l'extraction des règles d'association et les transmuter vers un contexte plus convenable au procédé utilisé.
- **Recherche d'Items Fréquents :** Cette étape consiste à extraire du contexte tous les ensembles d'attributs binaires appelés itemsets qui sont fréquents dans le contexte.
- **Génération des règles d'association :** On pourra générer les règles d'association en se servant des itemsets obtenus dans l'étape précédente.
- **Visualisation et interprétation :** Cette étape consiste à regrouper les connaissances, les interpréter et les visualiser à l'utilisateur.

e)-Les Techniques utilisés dans un Processus DataMining Distribué :

L'exécution d'un processus de Fouille de données en mode distribué nécessite souvent suivre l'une des approches suivantes [02M, 09, 06W]:

1)-Le Clustering distribué : il est utilisé pour aider deux ordinateurs à collaborer et former un cluster, il peut inclure au moins deux sites appelés aussi « nœuds ».

2)-La Classification distribuée : elle permet d'extraire automatiquement des connaissances à partir d'un ensemble de données en se servant d'outils statistique.

3)-Les Règles d'association distribuées : l'approche sur laquelle on va baser notre travail, elle sera notre objet d'étude.

I.3.5-Les Règles d'Association Distribuées :

A)-Définition :

L'extraction des règles d'association distribuées est une technique permettant de dévoiler des règles (formules) intelligibles et utilisables dans un ensemble de données volumineux pour exprimer des relations (associations) entre les items d'une base de données en mode réparti (distribué) [02M, 06W].

B)-Principe de Fonctionnement :

A la différence par rapport aux méthodes d'extraction de règles d'association classique, celles qui sont distribuées nécessitent des algorithmes plus spécialisés. Ces derniers aident à localiser et regrouper les règles d'association à partir d'un environnement distribué, nous expliquerons ces algorithmes avec plus de détail dans les chapitres qui vont suivre[02M, 06W].

Avant de connaitre les algorithmes dédiés à l'extraction des règles d'association distribués, on doit d'abord voir la rubrique suivante qui va donner un rappel sur les systèmes multi agents, leur fonctionnement et leurs rôles dans l'exécution distribuée du Data Mining.

I.4-Rappels sur les Systèmes multi Agents :

I.4.1-Les Agents :

1)-Définition :

Le mot « Agent » est un terme générique qui désigne tout programme informatique capable d'agir et de fonctionner d'une façon autonome, parfois doté de mobilité, parfois doté de fonctions d'apprentissage. Il adapte son comportement à son environnement et en mémorisant ses expériences , se comporte comme un sous-système capable d'apprentissage : Il enrichit le système qui l'utilise en ajoutant au cours du temps des fonctions automatiques de traitement , de contrôle , de mémorisation ou de transfert d'informations et en plus il engendre un aspect distribué à tout projet [10,17W].

2)-Caractéristiques des Agents :

- **L'autonomie :** L'agent doit pouvoir prendre des initiatives et agir sans intervention de l 'utilisateur. Dans le contexte du web il doit pouvoir agir si l'utilisateur est déconnecté.
- **Capacité à communiquer et à coopérer :** L'agent doit pouvoir échanger des informations avec d'autres agents, avec des serveurs et avec des humains.
- **Capacité à raisonner, à réagir à leur environnement :** L'agent doit être capable de s'adapter à son environnement et aux évolutions de celui-ci.
- **La mobilité :** Les agents doivent pouvoir être Multi-plate-forme et Multi-architecture. En cas de besoin, Ils doivent pouvoir se déplacer sur le réseau où ils accomplissent des tâches sans que l'utilisateur intervienne.

I.4.2-Les SMA (Systèmes Multi -Agents) :

a)-Définition :

Un système multi-agent (SMA) est un système composé d'un ensemble d'agents, situés dans un certain environnement et interagissant selon certaines relations. Un agent est une entité caractérisée par le fait qu'elle est au moins partiellement autonome. Ce peut être un processus, un robot, un être humain, etc.

Les systèmes multi-agents répondent aux besoins d'applications complexes et distribuées. L'objectif est de permettre d'acquérir les concepts fondamentaux du domaine des SMA et d'appréhender la dimension collective de la résolution, de situer ce domaine par rapport à des domaines connexes et aborder leur mise en œuvre [10,11,17W].

b)-Typologie des Agents dans un SMA :

Agent Réactif : possède des connaissances sur sa tâche et son environnement, non sensible aux changements, capable de communiquer et d'agir.

Agent Cognitif : sensible aux changements, capable de faire des décisions, doté de mécanisme d'apprentissage.

Agent Hybride : la combinaison des deux types précédents.

La prochaine section va expliquer le concept de fusion entre la technologie du Data Mining et celle des Systèmes Multi Agents, le rôle de chaque discipline par rapport à l'autre et le résultat de cette combinaison de technologies.

I.5-La Fusion « Data Mining-SMA » :

I.5.1-Présentation :

La technologie de l'exploration de données est un moyen d'identifier des modèles et des tendances de grandes quantités des données. Elle adopte souvent une méthode d'intégration de données pour générer des entrepôts de données, sur lesquels sont rassemblées toutes les données dans un site central, puis exécutez un algorithme sur ces données pour en extraire l'utile Prédiction ou une évaluation des connaissances [12,13].

Cependant, les Techniques de Data Mining impliquant aussi des environnements complexes pour faire face aux changements pouvant affecter l'ensemble de la performance du système. Les Systèmes multi-agents (SMA) ont souvent affaire à des applications complexes qui nécessitent la résolution de problèmes distribués. Dans de nombreuses applications, le comportement individuel et collectif des agents dépend des données observées à partir des sources distribuées [12,13].

La Distribution de la fouille de données est originaire de la nécessité de fouille sur les sources de données décentralisées .Le domaine de la Distribution du Data Mining(DDM) traite ces défis dans l'analyse des données distribuées et offre de nombreuses solutions algorithmiques pour effectuer différentes analyses de données et les Processus Data Mining dans une approche fondamentalement distribuée de manière à accorder une attention particulière aux contraintes de ressources .Depuis que les systèmes multi-agents sont conçus, les agents ont des caractéristiques proactives et réactives qui sont très utiles pour la connaissance des Systèmes de gestion, la combinaison des DDM avec MAS pour les applications de données intensives est séduisante [12,13].

I.5.2-Rôle des agents dans le Data Mining :

Dans le Data Mining basé agents, un agent est une entité logicielle caractérisée par les capacités suivantes [12,13] :

1) Situer les sources de données, les quantifier et vérifier leurs homogénéité.

2) Interagir avec la source de données (et/ou) d'autres agents,

3) Recevoir, collecter des données brutes,

4) Traiter les données d'une source ou plusieurs sources de données lui permettant de produire des connaissances,

5) Coopérer avec les autres agents pour produire de la connaissance pertinente et utile.

I.5.3-L'intérêt de la contribution agents – mining :

Un système Data Mining basé agents contribue positivement au processus de Data Mining sur plusieurs aspects :

D'abord, un système Data Mining Basé sur les Agents (DMBA) garantit le parallélisme qui améliore la vitesse, la performance et la précision. La nature distribuée d'un système d'agents autorise l'exécution parallèle de processus Data Mining sans se soucier du nombre de sources de données [12,13].

En Deuxième lieu, le paradigme des agents fournit aux utilisateurs d'un système de Data Mining la capacité de suivre progressivement le processus de découverte des connaissances durant les différentes étapes. par exemple, un utilisateur peut vouloir visualiser le modèle de connaissance obtenu par un agent particulier avant que l'intégration n'ait lieu [12,13].

Un autre avantage est adopté par le système DMBA, c'est la faculté des agents à rassembler, chercher et traiter de l'information hors d'une source de données. Ceci est possible grâce à l'attribut de la mobilité d'un agent logiciel, ces types d'agents peuvent être :

I.5.3.1-Les Agents Collecteurs :

Ce type d'agent s'occupe d'exécuter la phase de collecte des données à partir d'une ou plusieurs sources de données.

Sur un système mono source, on aura au maximum un seul agent de ce type. Dans le cas d'un système multi sources de données, la fonction de ces agents dépend du nombre de sources de données et de l'homogénéité/hétérogénéité des données, car dans le cas d'hétérogénéité l'agent collecteur doit pouvoir s'adapter au type de données à rassembler [12,13]..

I.5.3.2-Les Agents Facilitateurs :

L'agent facilitateur n'existe que lorsqu'il y a le cas de plusieurs agents collecteurs ou agents mineurs, sa fonction est soit de rassembler les résultats obtenus avec les agents mineurs, soit de rassembler les données des agents collecteurs pour appliquer son filtrage [12,13]..

I.5.3.3-Les Agents Mineurs :

Ce type d'agent pouvant se déplacer ou non, il applique sa propre technique de fouille de données sur un ensemble de données (parfois obtenus à partir d'agents collecteurs) et retourne son modèle final à un point central qui va décider du résultat de tous les agents mineurs [12,13].

Le paragraphe qui va suivre expliquera les problèmes et les contraintes rencontrés en essayant de combiner les deux disciplines et la difficulté d'un tel travail.

I.6-Problèmes Liés au Data Mining basé agents :

Bien qu'il offre le moyen de faire du Data Mining distribué, le Data Mining basé sur les agents n'est pas épargné des problèmes spécifiques aux techniques du Data Mining, tel que les données manquantes, et le problème de scalabilité (extensibilité et adaptabilité).

En outre, les systèmes Data Mining basés agents dans beaucoup de cas sont plus difficiles à concevoir et à implémenter que les systèmes conventionnels de Data Mining vu les caractéristiques spécifiques des agents et la distribution des tâches.

La section prochaine décrit des travaux incluant la technologie du Data Mining combinée au système d'agents, c'est des projets connus sur le marché et sur le Net. Pour aboutir à notre objectif final, il nous faut étudier ces différents travaux, les prendre comme témoin et d'en effectuer une comparaison de performance [12,13].

I.7-Quelques systèmes existants du Data Mining basé agents :

I.7.1-Le Système MAD-IDS(ADMI) (Intrusion Detection System) 2011 :

a)Présentation :

Un système qui aide à surveiller les événements qui se produisent dans un système distribué d'ordinateurs ou d'un réseau et analyse les signes d'intrusions est connu comme le système de détection d'intrusion (IDS). Les IDS doivent être dotés de précision, d'adaptation et extensibilité. Bien que bon nombre des techniques établies et de produits commerciaux existent, leur efficacité laisse place à l'amélioration.

Un grand nombre de recherches ont été menées sur la détection de l'intrusion dans un environnement distribué pour pallier les inconvénients de l'approche centralisée. Cependant, l'IDS distribué souffre d'un certain nombre d'inconvénients par exemple, des taux élevés de faux positifs, la faiblesse de l'efficacité…etc.

Dans cet exemple, Le système présenté repose sur un ensemble d'agents intelligents qui recueillent et analysent les connexions réseau et les techniques de data mining sont avérés utiles pour détecter les intrusions, Effectuer des expertises, ce qui a montré des performances supérieures des IDS distribués par rapport à aux systèmes de détection décentralisés classiques [14,13W,17W].

b)Le principe de fonctionnement:

Fondamentalement, le système de décompose en deux IDS principales qui peuvent être distinguées : La détection d'anomalies et l'utilisation abusive. L'objectif du projet est de jouer les deux rôles décrits précédemment (détection d'anomalies, détection d'abus) à la fois, et le principe de fonctionnement du système est expliqué comme suit [14,13W,17W] :

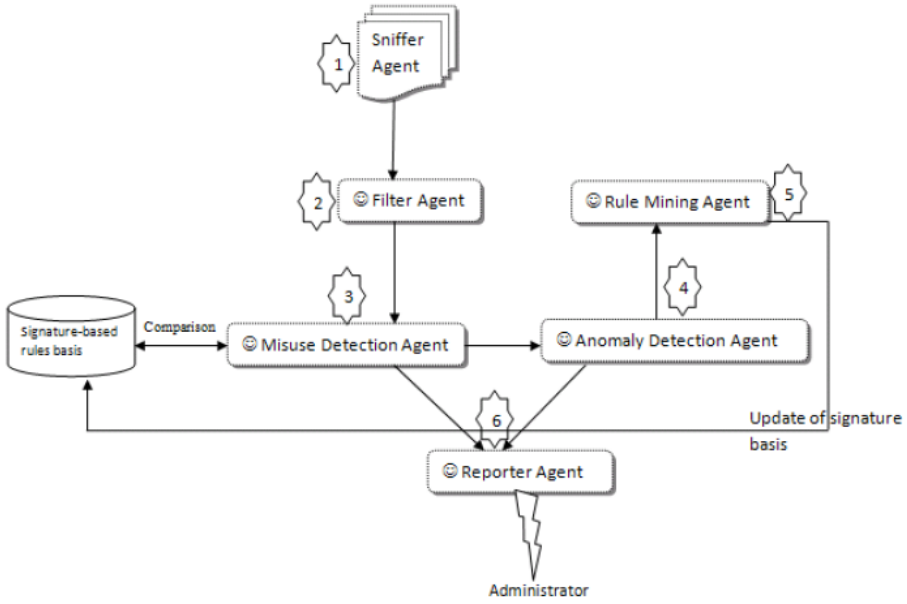


Figure3 : L'architecture générale du système MAD-IDS

1 : Sniffer Agent est un agent qui joue le rôle d'éclaireur, c'est lui qui lance la connexion à la source de données (locale ou distante) et rassemble des informations concernant une transaction prédéfinie.

2 : Les paquets sont vérifiés par le Filter Agent.

3 : Les paquets jugés correctes sont passé à l'Agent détecteur d'abus (Amusus Detection Agent)

Dans le cas de test positif, les paquets victimes d'abus sont marqués et l'adresse de l'agresseur est retenue par l'agent de détection d'abus.

4 : les paquets jugés abimés ou anormaux sont examinés par l'agent de détection des anomalies

(Anomalies detection Agent) qui va exécuter un algorithme de clustering afin de déterminer

le type d'anomalies et l'adresse d'où vient l'intervention malveillante.

5 : le Rule Mining Agent est un agent qui récupérer le résultat du détecteur d'abus et du détecteur d'anomalies aussi pur appliquer l'algorithme d'extraction de règles qui va aider à cerner les sites malveillants selon l'abus ou l'anomalie trouvé et composer une carte représentant la situation actuelle du réseau et la mise à jour de la table des signatures des agressions connues.

6 : Le Reporter Agent va rédiger un journal ou un rapport de détection et un aperçu sur la situation pour le présenter à l'administrateur (l'utilisateur) du réseau ou à celui qui a lancé le processus de détection.

I.7.2-Le système PAPYRUS 1999 :

a)-Présentation :

L'exploration de données est un problème pour le quelle le clustering fournit une alternative compétitive aux ordinateurs spécialisés de haute performance pour les grands ensembles de données distribuées. Des Nœuds (grappes, points) fournissent une infrastructure naturelle pour les grands ensembles de données distribués. Ces nœuds peuvent être connectés par des réseaux de matières premières pour former ce que nous appelons méta-clusters et des réseaux de haute performance pour former ce que nous appelons des super-amas [07,18,07W,08W].

Papyrus est conçu pour les données des processus de fouille réparties sur des clusters, des méta-clusters, et super-amas. Nous décrivons dans un paragraphe le principe de fonctionnement de ce système.

B)- principe de fonctionnement:

Papyrus est un système de type Standard MADM (Multi Agents DataMining avec architecture distribuées standard) constitué de couches d'outils logiciels et de services réseau dédiés aux systèmes distribués pour l'exploration de données et le calcul intensif. Les applications Papyrus peuvent être conçues pour accéder à l'une des quatre couches suivantes, selon les besoins.

Les différentes techniques de datamining utilisent des stratégies possibles, ces dernières dépendent des données elles-mêmes, de la distribution appliquée, des ressources disponibles et des occurrences requises tel que [07,18,07W,08W] :

MR : Move Result, la possibilité de déplacer l'ensemble des résultats volumineux et extensibles d'un processus de fouille locale vers un site central d'un réseau.

MM : Move Model, la capacité à déplacer des modèles prédictifs d'un site vers un autre dans un réseau.

MD : Move Data, la possibilité de déplacer des volumes gigantesque de données à traiter d'un site à un autre.

Le but principal du système est de réduire la charge aux applications qui peuvent déplacer des données (MD) d'un nœud à l'autre à l'aide de la couche inférieure. L'application des modèles de déplacement (MM) et les résultats (MR) des nœuds obtenus sont obtenus à l'aide de la couche supérieure.

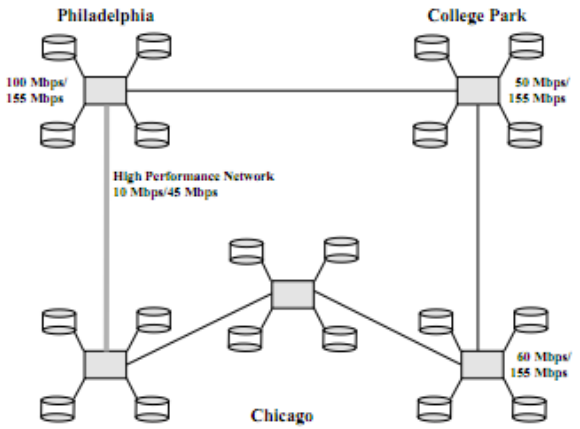


Figure 4 : Un cluster de stations de travail obtenu avec Papyrus

Les autres systèmes se concentrent sur l'application du MR, MM mais le système Papyrus s'occupe d'appliquer les trois opérations au niveau d'un réseau dans un cadre distribué à base d'agents pour bien alléger la transmission afin, par conséquent on pourra lancer d'autres processus sans craindre la saturation du réseau [18,07W].

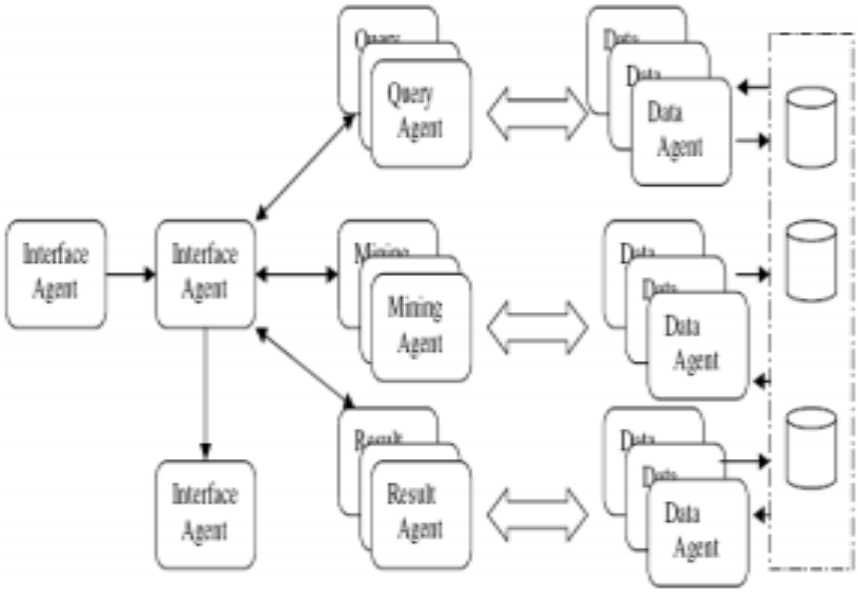


Figure 5 : Architecture générale d'un Système MADM

I.7.3-Le système PADMA 1997 :

a)-Présentation :

PADMA (Parallel Data Mining Agents) est un système Data Mining basé agents DMBA conçu pour répondre au problème d'échelle (scaling problem) de Data Mining, un tel système a été décrit la première fois par Mr H.Kargupta et I.Hamza oglu en 1997 dans le cadre d'un projet intitulé « scalable distributed data mining agent based application ».

Les concepteurs PADMA suggèrent que la nature très distribuée des données et des calculs des environnements Informatiques jouera un rôle important dans la conception de la prochaine génération des systèmes de Data Mining basés agents [18,3,11W,12W].

PADMA se divise en trois parties qui sont :

- Agents Data Mining,
- Un facilitateur pour coordonner entre agents,
- Une interface utilisateur.

Les agents Data Mining ont accès direct aux données pour en extraire de la connaissance, et donc chaque agent a besoin de se spécialiser dans un domaine particulier des données de la source avec laquelle il est rattaché.

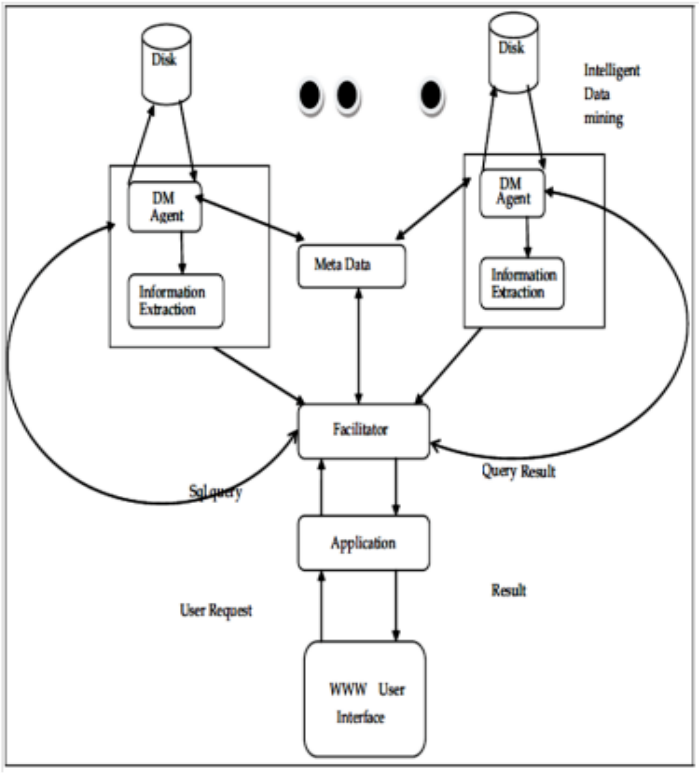


Figure 6 : Architecture du Projet PADMA

b)-Principe de fonctionnement :

Chaque agent a son propre sous-système de disque et exécute des opérations d'entrées sorties sur les données indépendamment des autres agents : c'est la clé vers l'exécution parallèle dans PADMA. Ainsi, les agents peuvent utiliser des techniques d'optimisation des entrées sorties locales pour augmenter leur vitesse de traitement.

Les agents partagent les connaissances extraites à travers l'agent facilitateur. Ce dernier présente le résultat du processus Data Mining à l'interface de l'utilisateur et achemine des réactions de l'utilisateur (feedbacks) vers les agents Data Mining.

PADMA résout le problème d'échelle « scaling problem » en réduisant le volume de communication inter-agents pendant le processus Data Mining [07W,08W].

I.7.4-Le système JAM :

a)-Présentation :

JAM (Java Agent for Meta-learning over distributed data bases) les auteurs de JAM ont identifié le besoin d'adapter au mettre à l'échelle (scalling) les algorithmes data mining aux données très volumineuses. Ce système est motivé par la prise en charge des données fondamentalement distribuées [07,18,09W,10W].

Le système JAM est une collection de programmes Data Mining liés par un réseau de sites. Chaque site consiste :

- Une base de données locale
- Un ou plusieurs agents Data Mining de base (base-learning agents)
- Un meta-data mining agent (meta learning agent)
- Un fichier de configuration utilisateur local
- une interface graphique utilisateur.
- Un Learning agent est un programme Data Mining de classification

b)-Principe de fonctionnement :

Les agents Data Mining de base calculent, dans un premier temps, des classifieurs de base d'une collection de base de données distribuées d'une manière parallèle , puis les agents meta-data mining intègrent ces classifieurs de base produits localement au niveau des différents sites. De plus, JAM a un module central indépendant, appelé CFM (Configuration File Manager) qui mémorise l'état actuel du système distribué et enregistre une liste de liste de sites participants au processus Data Mining distribué [18, 10W].

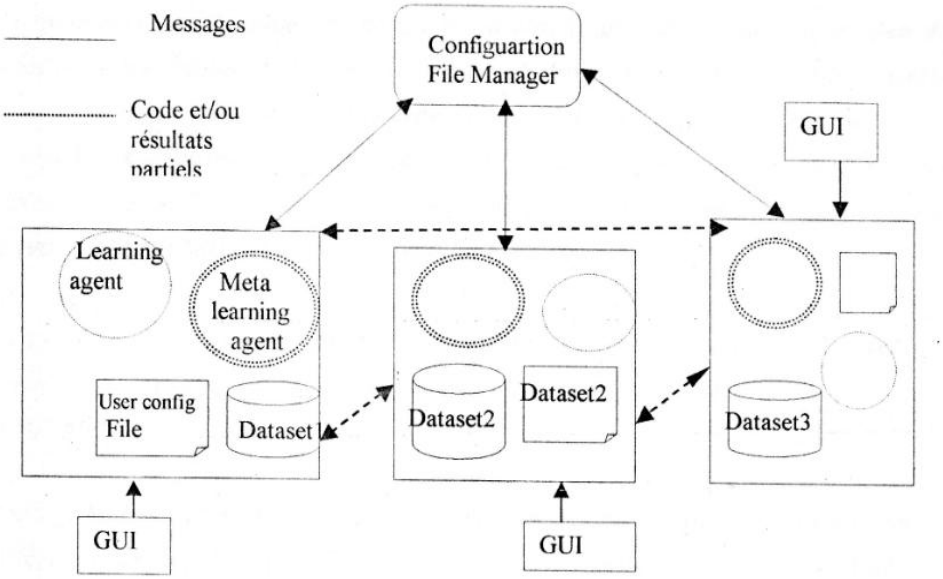


Figure 7 : Architecture JAM avec 3 sites de données

On note que sur chaque site de données , l'agent Data Mining local opère sur la base de données locale pour calculer le classifieur de base et peut importer des classifieur depuis d'autres sites et les combiner avec son propre classifieur local en utilisant l'agent meta-data mining local. JAM résout le problème d'échelle (scaling problem) de Data Mining en calculant un meta classifieur qui intègre tous les classifieurs de base.

Le CFM assume un rôle passif pour l'entretien de la configuration du système. Il maintient une liste de sites actifs qui coordonnent l'activité du meta Data Mining.

JAM implémente le CFM et les « learning agents » par des programmes java multi threads alors que les « meta- learning agents » sont implémentés par des applets java par ce qu'ils peuvent migrer d'un site à l'autre.

Les membres de l'équipe ont conçu initialement JAM dans une application de découverte de fraudes dans les systèmes d'informations des banques. Cette expérimentation utilise le système pour détecter les fraudes dans les transactions effectuées par les cartes de crédit.

Le système n'exige pas le transfert de données à travers les différents sites, ce qui garantit la sécurité des données des sites participants, les différents sites peuvent choisir des programmes Data Mining différents (De classification).

Dans la section suivante, on va désigner des critères pour permettre de faire une étude comparative des performances concernant les travaux décrits précédemment.

I.8-Etude comparative des systèmes DMBA :

a)-Les Critères de comparaison :

Pour pouvoir comparer les systèmes de Data Mining basé agents présentés précédemment, on se donne l'ensemble des critères suivant :

1) Tâches Data Mining supportées par le système ?
2) Communication entre agents pendant au après le processus Data Mining (pendant au après l'extraction des connaissances) ?
3) Est-ce que le système peut réutiliser des algorithmes Data Mining existants sans modification ?
4) Qui fait le choix de la technique et l'algorithme réalisant une tache données
5) Le système est –il interactif ?
6) Le type de sources de données ?
7) Le type de symbiose entre les deux technologies (Agents - DataMining) ?
8) Le type de synchronisation appliqué entre les agents ?

Le Tableau 1 répond aux questions concernant les critères pour chaque projet en ce qui concerne l'interactivité, la distribution des tâches, la réutilisabilité et le nombre de sources de données manipulablées. Ensuite, on pourra choisir l'architecture la plus adaptées à notre système.

b) Tableau comparatif des systèmes étudiés :

N°	Le Critère étudié	PADMA	JAM	PAPYRUS	MAD-IDS
1	***Tâches supportées par le système ?***	Analyse des clusters	classification	clustering	Clustering Règles -assoc.
2	***Communication avant, pendant, après le processus data mining ?***	Au Début et fin Du processus	Après le processus data mining	Après Chaque étape Du processus	Au Début et fin Du processus
3	***Réutilisation des algorithmes de data mining ?***	oui	oui	oui	non
4	***Qui fait le choix de la technique et l'algorithme réalisant une tache data mining ?***	La technique et l'algorithme sont prédéterminés	L'utilisateur via le fichier de configuration utilisateur	La technique et l'algorithme sont adaptatifs	La technique et l'algorithme sont prédéterminés
5	***Interactivité avec l'utilisateur ?***	oui	Oui	oui	oui
6	***Le type de sources de données ?***	Nbre Dataset prédéfinit	3 Dataset prédéfinit + fichier config	Multi-Sources (4 supports)	Multi-sources Web
7	***Le type de symbiose entre les deux technologies (Agents - DataMining) ?***	coopération	coopération	Contribution des Agents	Contribution des Agents
8	***Type de Synchronisation ?*** ***(nombre d'agents qui pouvent interagir simultanément)***	+eurs Grace à L'Agent Facilitateur	1	1	7 agents

Tableau1 : Tableau comparatif des systèmes DMBA candidats

I.9-Discussion et synthèse :

Après avoir analysé le tableau de comparaison entre les quatre projets sélectionnés, on déduit les éléments suivants :

- Le projet MAD-IDS est basé sur le clustering et l'extraction des règles d'association, permet une interactivité avec l'utilisateur et jusqu'à 7 agents peuvent fonctionner de manière synchronisée et autonome, mais son architecture fait de lui un produit figé dans son domaine d'application (non réutilisable).
- Le projet PAPYRUS possède une architecture intéressante, Data mining basé sur la clustering distribué, mais la synchronisation entre les agents lui fait défaut, ce qui l'exclue de notre choix.
- Le système JAM possède aussi une architecture fortement distribuée, Data mining orienté vers la classification automatique, une coopération parfaite entre la technologie de Datamining et les SMA, toutefois il souffre du même problème de synchronisation et scalabilité entre agents.
- Le système PADMA possède une architecture standard de système Data Mining distribué, basé sur les agents, il possède une interactivité forte avec les utilisateurs, une réutilisabilité des modules et le problème de synchronisation ne se présente pas.

Le choix de l'architecture de base pour notre projet se fera selon les deux critères les plus importants dans le tableau précédent (ligne 7 et 8), le premier est le nombre d'agents pouvant être impliqués dans le processus de fouille de données, car plus le système peut inclure une quantité nombreuse d'agent plus l'aspect distribué et l'extensibilité seront mieux garanties par le système envisagé, le deuxième critère est celui de l'autonomie et la coopération entre les agents, dans ce même contexte, plus les agents sont autonomes, plus cela donne des performances au système et le même avantage apparait si les agents sont parfaitement coopérants entre eux.

Selon ces deux critères favorisés, l'architecture PADMA semble être la plus appropriée pour servir d'architecture de base à notre travail. Du point de vue extensibilité et aspect distribué du système, cette plateforme répond aux conditions requises, car elle peut inclure un nombre indéfini d'agents .

Le travail demandé consiste à concevoir une plateforme adoptant ce modèle d'architecture et de lui attribuer des améliorations, maintenant il faut affronter les problèmes liés à la réalisation d'une telle architecture.

Conclusion :

L'étude de l'intégration et l'interaction des disciplines des agents et de la fouille de données en mode distribué apporte plusieurs avantages dans l'amélioration des traitements dans les processus d'apprentissage. Toutefois cette fusion rencontre des problèmes que les scientifiques cherchent à surmonter pour le bénéfice des deux disciplines.

On a essayé de faire un tour d'horizon sur les notions évoquées par ces deux disciplines et leur fusion ultime dans un aspect résumé afin de ne pas s'approfondir sur un point par rapport aux autres puisque c'est un domaine très vaste et nécessite des ressources et un temps considérable.

Le chapitre suivant intitulé « Algorithmes Distribués » va donner une explication sur les algorithmes utilisés dans l'extraction des règles d'association plus en détail et les modifications qu'on va apporter au projet sélectionné (PADMA) et le choix de l'algorithme RAD le plus adopté à notre travail (PADMA-RAD).

Algorithmes Distribués pour L'extraction des règles d'association

-Introduction

-Les Algorithmes Distribués

-Les Règles d'Association Distribuées « RAD »

-Récapitulatif des Algorithmes RAD

-Discussion et synthèse

-Conclusion

Chapitre 02 : Algorithmes Distribués

Pour L'Extraction des Règles d'Association

2.1-Introduction :

Beaucoup de travaux de recherche ont été menés sur la fouille de données dans le cadre de l'extraction des règles d'association distribuées. Cette technique permet la découverte de règles intelligibles et exploitable dans un ensemble de données volumineux, règles exprimant des associations entre items ou attributs dans une base de données distribuée.

De nos jours, les données sont distribuées sur plusieurs sites, il devient alors primordial de mobiliser d'importants moyens afin d'extraire des informations. Pour répondre à ces besoins, le Data Mining distribué est apparu proposant une multitude de techniques connues pour leurs facultés à extraire des informations dans ces méga bases. Parmi les techniques les plus répondues, nous nous intéressons à la technique des règles d'association distribuées. Cette dernière permet de découvrir les relations significatives entre les attributs en produisant des règles d'association destinées à être utilisées dans un but décisionnel, voire organisationnel.

Dans chapitre on va présenter certains algorithmes de bases dédié à la recherche des règles d'association dans un environnement purement distribué, des algorithmes comme « Count Distribution CD », « Data Distribution DD » ensuite les différents algorithmes de recherche des règles d'association distribuées qui utilisent la tentative d'équilibrage de charge, comme « Intelligent Data Distribution IDD », « Hash Partioned Apriori HPA » et l'algorithme « Common Condidate Partitionned Database CCPD ».

2.2-Les Algorithmes Distribués :

Il existe plusieurs algorithmes pouvant être utilisés dans une approche distribués pour un processus datamining, dont chacun est choisi selon le but qu'il remplit et la manière avec laquelle il procède, citons [13W, 01M, 02M,16, 13, 07, 03] :

Les Arbres de Décisions : cet algorithme consiste à se service d'un critère choisis pour diviser un ensemble d'éléments en N sous-ensembles.

Algorithme RBM (Raisonnement basé sur le mémoire) : elle exige des ressources, notamment une zone mémoire importante et consiste à appliquer un traitement de recherche du plus proche voisin PPV.

Algorithme Neuronal (Réseaux de neurones) : basé sur un concept biologique, il consiste à imiter le cerveau humain afin d'accomplir des manipulations sur les données de manière numérique, non pas symbolique.

Algorithme Génétique : une méthode heuristique d'optimisation qui s'appuie sur le concept de la génétique et celui de l'évolution naturelle.

Les Règles d'association : comme son nom le décrit, cet algorithme consiste à rechercher des associations, c'est-à-dire cette méthode permet de découvrir des connaissance en parcourant les association et les liaisons existantes entres les données.

Rappel sur les Techniques du Datamining Distribué DMD :

Les différentes méthodologies utilisées dans un processus DMD sont [13W, 01M, 02M, 13, 07, 03] :

- La Classification distribuée : la classification permet d'extraire automatiquement des connaissances depuis un ensemble de données, il existe plusieurs méthodes pour l'effectuer dont la classification distribuée est l'une d'elles.

- La Recherche des Règles d'Association Distribuées : une technique pour trouver des relations entre des objets dans un ensemble de données non centralisées (dans ce cas).

- Le Clustering Distribué : c'est une technique utilisée pour mette en commun la puissance de deux ordinateurs, elle comprend au moins deux sites appelés « nœuds ».

2.3-Les Règles d'Association Distribuées :

2.3.1-Définition :

C'est une technique qui permet de dévoiler des règles (formules) intelligibles utilisables dans un ensemble de données très volumineux pour exprimer des associations (relations) entres des items d'une base de données.

A la différence de la méthode d'extraction des règles d'association classique pour un ensemble de données centralisé, celles qui sont distribuées nécessitent des algorithmes spécialisés dans les systèmes distribués [16,02M,06W,13W].

2.3.2-Les Algorithmes de RAD :

Des algorithmes spécialisés permettant de regrouper et de localiser des règles d'association à partir d'un environnement distribué (non centralisé) , dans l'ouvrage en cours nous présenterons quelques algorithmes de recherche de RAD les plus connus.

Count Distribution CD, Data Distribution DD, Hybride distribution HD, Intelligent Data Distribution IDD…

Dans les premiers travaux sur l'équilibrage de charge, la plupart des solutions proposées étaient statiques. Les algorithmes statiques nécessitent une connaissance apriori des tâches et du système sur lequel elles vont être exécutées. Par conséquent ces algorithmes ne conviennent pas pour les applications et les systèmes ayant un comportement imprévisibles.

L'intérêt qui leur est porté se justifie par la popularité des multi-ordinateurs caractérisés par une très grande hétérogénéité mais aussi par la complexité des tâches qui y sont souvent exécutées. Les algorithmes dynamiques peuvent être distribués ou centralisés, mais avec les systèmes massivement parallèles, largement utilisés aujourd'hui pour le traitement parallèle, un équilibrage de charge distribué devient une exigence pour la scalabilité d'une part mais aussi pour éviter le goulot d'étranglement de l'approche centralisée d'autre part. Ainsi, cette section sera consacrée à l'étude de quelques algorithmes d'équilibrage de charge dynamiques[16,02M,13W].

En se basant sur l'algorithme APRIORI, plusieurs méthodes algorithmiques prévues pour l'extraction des règles d'association distribuées ont vu le jour. On a en première génération le DD (Data distribution) et le CD (Count distribution), ces deux algorithmes vont servir de base pour d'autres plus complexes et plus performants.

A)-L'algorithme Count Distribution CD :

Proposé par R. Agrawal et J.C Shafer en 1996, cet algorithme basé sur l'algorithme APRIORI, est réalisé dans un environnement distribué, où chaque site traite sa portion locale de la base de transactions, calcule les supports locaux des candidats et les diffuse aux autres sites pour calculer le support global [16,02M,13W].

Algorithme Count Distribution « CD » :

1. Partitionnement horizontal des données dans chaque site.
2. Pour i=1 , chaque site ;
3. Génère l'ensemble des candidats de taille 1, noté C1 ;
4. Calcule le support local de l'ensemble des candidats C1 ;
5. Diffusion des supports locaux aux autres sites ;
6. Détermine l'ensemble des motifs fréquent de taille 1, noté L1 ;
7. Répéter
8. I=i+1 ;
9. Générer l'ensemble des candidats Ci à partir de Li-1 ;
10. Calculer le support local des Candidats Ci ;
11. Diffuser les supports locaux aux autres sites ;
12. Déterminer l'ensemble des motifs fréquents de taille i, noté Li+1 ;
13. Jusqu'à ce que tous les motifs fréquents soient trouvés ;

Inconvénients et Avantages de l'Algorithme CD :

Le principe de l'algorithme est simple mais il gagne en complexité avec la taille de la base de données, le nombre d'agents générés doit garantir l'accès à l'information sur les différents sites mais cela devient de plus en plus couteux en ressources, en temps et va poser un problème de perte de synchronique entre les agents, car la communication entre eux sera exponentielle, parfois même il y a des risques de déséquilibrage de charges

B)-L'algorithme Data Distribution DD :

L'algorithme Data Distribution basé sur l'algorithme Apriori, est proposé par R.Agrawal et J.C. Shafer en 1996, il diffère de Count Distribution dans le sens où chaque site traite un ensemble de candidats différents, ce qui impose d'accéder aux portions de la base de transaction des autres sites [16,02M,13W].

Algorithme Data Distribution « DD » :

1. Partitionnement vertical des données dans chaque site.
2. Pour i=1, chaque site :
3. Déterminer les candidats locaux de taille 1, noté C1 ;
4. Calcule les supports des candidats locaux à partir des transactions locales et distantes
5. Détermine les motifs fréquents locaux de taille 1, noté L1 ;
6. Diffuse l'ensemble des motifs fréquents locaux L1 aux autres sites ;
7. Détermine l'ensemble des motifs fréquents globaux Li ;
8. Répéter
9. I=i+1 ;
10. Générer l'ensemble des candidats C1 à partir de Li-1 ;
11. Calculer les supports des candidats locaux à partir des transactions locales et distantes ;
12. Déterminer les motifs fréquents locaux de taille i, noté Li ;
13. Diffuser l'ensemble des motifs fréquents locaux Li aux autres sites ;
14. Déterminer l'ensemble des motifs fréquents globaux Li ;
15. Jusqu'à ce que tous les motifs fréquents soient trouvés ;

Inconvénients et Avantages de l'Algorithme DD :

Cet algorithme s'avère meilleur que Count Distribution lorsque la base contient beaucoup de motifs (itemset) distincts et lorsqu'on choisit un support minimal peu élevé. Cependant d'autres inconvénients se posent tel que le nombre important de données échangées par rapport à l'algorithme Count Distribution ; c'est-à-dire que cet algorithme reste aussi limité par un seuil concernant la taille de la base do données et le nombre de sites à visiter, ce seuil qui dépasse celui de l'algorithme CD.

C)-L'algorithme Intelligent Data Distribution IDD :

Cet algorithme est une amélioration de l'algorithme Data Distribution « DD », il est venu pour remédier aux problèmes posés par le modèle de communication de l'algorithme DD qui utilise une communication all-to-all, en utilisant une topologie d'anneau virtuel ; de plus, le mode de partitionnement des candidats de l'algorithme IDD est différent de celui de l'algorithme DD.

En effet, au lieu d'un partitionnement Round Robin de candidats, un partitionnement intelligent est réalisé selon les items préfix, c'est-à-dire les candidats avec le même item préfix sont mis dans la même partition. Par conséquent, seules les transactions qui contiennent les items candidats seront consultées. Ce qui réduit significativement le nombre d'accès aux BDD distantes [16,02M,13W].

Inconvénients et Avantages de L'algorithme Intelligent Data Distribution IDD :

Dans le contexte où cet algorithme IDD (Intelligent Data distribution) vient résoudre le problème de communication entre les agents, il aide à faire gagner du temps aux agents dans leurs traitements. Il aide à diminuer le volume de données à échanger, cela implique aussi qu'il peut engendrer des pertes d'information si la base de données est très volumineuse.

D'un autre côté, la phase de partitionnement aide les agents à se partager les données au début du traitement, cela permet de gagner un temps considérable en abrégeant la communication pendant le traitement des agents, mais dans certains cas cette phase s'avère inutile surtout si les bases de données typiques (des domaines d'application précis) .

Du point de vue implémentation, les deux caractéristiques de cet algorithme (partitionnement intelligent, structure en anneau virtuel) font de lui une méthode très simple à concevoir et à réaliser, mais l'aspect dynamique de cet algorithme reste à améliorer.

D)-L'algorithme Hash Partioned Apriori HPA :

L'algorithme HPA est conçu dans le même esprit que l'algorithme IDD.Il partitionne les itemsets candidats entre les différents processeurs à l'aide d'une fonction de hachage. Ce partitionnement permet de résoudre les problèmes liés au débordement de la mémoire lorsque le nombre d'itemsets candidats est important. Il réduit aussi le nombre de communications puisque toutes les transactions de la base de données ne sont plus diffusées [02M,13W].

L'algorithme HPA a été implémenté sur un cluster de PC's et chaque nœud procède comme suit à l'itération k :

1. Génération des k-itemsets candidats : chaque processeur génère les k-itemsets candidats en utilisant les (k-1) – itemsets fréquents de l'étape précédente. La fonction de hachage est ensuite appliquée aux itemsets candidats pour déterminer l'ID (identifiant) de leur processeur hôte. Chaque itemset candidat est ensuite inséré dans la table de hachage de son processeur hôte.
2. Calcul du support : chaque processeur génère l'ensemble des k-itemsets pour les transactions stockées sur son disque local en éliminant ceux dont le support est inférieur à minsup. La fonction de hachage utilisée dans la phase 1 est alors appliquée à chaque k-itemset pour déterminer l'ID de son processeur hôte. Chaque processeur est ensuite responsable d'incrémenter la valeur du support pour les itemsets générés localement et ceux envoyés par les autres processeurs.
3. Déterminer des itemsets fréquents : une fois que toutes les transactions sont traitées, chaque processeur détermine localement les itemsets fréquents à partir de ses itemsets candidats. La totalité des itemsets fréquents est obtenue en sommant les itemsets fréquents provenant de tous les processeurs.

L'algorithme Common Candidate Partitioned Database CCPD :

L'algorithme CCPD est une implémentation de l'algorithme Apriori sur une architecture à mémoire partagée. Avec l'algorithme CCPD, la base de données est logiquement partitionnée entre les différents processeurs qui utilisent un arbre de hachage commun des candidats. La construction de l'arbre de hachage est parallélisée. Ainsi chaque processeur génère un sous ensemble disjoint de candidats [02M,13W].

Pour garantir la cohérence des données, un verrou est associé à chaque feuille de l'arbre. Lorsqu'un processeur veut insérer un candidat dans l'arbre, il commence à la racine et applique successivement le hachage sur les items constituant l'itemset candidat jusqu'à atteindre une feuille. Ensuite, le processeur acquière le verrou associé à la feuille et insère le candidat. Pour le calcul des supports, chaque processeur compte la fréquence des itemsets de la base de données locale et le processeur maitre sélectionne ensuite les itemsets fréquents [16,02M,13W].

2.3.3-Récapitulatif des Algorithmes de RAD :

Les algorithmes d'extraction des règles d'association distribuées forment un ensemble riche et polyvalent. La figure 8 donne un récapitulatif de l'évolution des algorithmes qu'on vient de décrire dans ce chapitre.

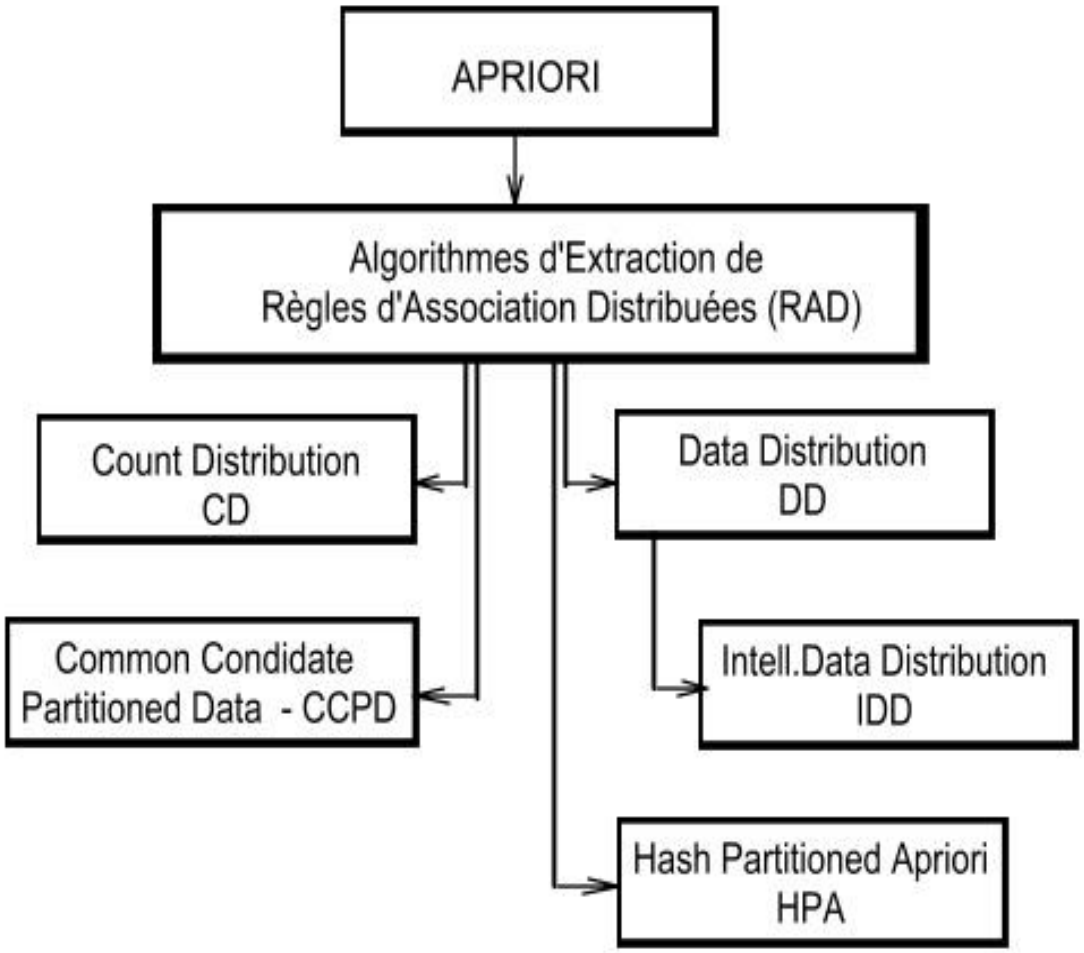


Figure 8 : Récapitulatif des Algorithmes Distribués.

Discussion et Synthèse :

Comme il a été déjà mentionné, l'algorithme A-PRIORI est la base de plusieurs familles d'algorithmes ayant pour but l'extraction des règles d'association distribuées.

La Famille des Algorithmes « Count distribution » propose un partitionnement horizontal et une distribution des tâches où chaque agent de Data Mining effectue son travail sur une portion de données locale et la communication ne se passe qu'à la fin, ce qui pose problème d'interactivité et de synchronisation surtout quand le volume de données est très important.

Le CCPD propose la même approche que le CD mais en utilisant une mémoire partagée et un hachage parallélisé des données qui va aider les agents à partager leurs informations, tout en évitant les redondances, c'est une approche intéressante.

L'algorithme « Data Distribution » propose un partitionnement vertical des données et une distribution des données où chaque agent se doit de traiter un ensemble distinct encourageant ainsi les agents à communiquer leurs informations. Cela répond aux problèmes posés avec le « Count Distribution », mais offre un mécanisme de communication qui peut s'altérer si la fréquence des transmissions devient importante.

L'algorithme « Intelligent Data distribution » propose la même chose que le « Data Distribution », pour répondre aux problèmes posés par les algorithmes précédents. Il propose une phase initiale qui consiste à appliquer un partitionnement intelligent des données ; en plus de cela, il offre une simplicité d'implémentation.

L'algorithme HPA propose la même chose que l'IDD mais il offre une fonction de Hachage au lieu d'un mécanisme de partitionnement intelligent. Il vient résoudre le problème de débordement de mémoire et encourage son utilisation sur des Cluster's de PC.

Dans notre cas, l'algorithme IDD a présenté plusieurs avantages. Grâce à ses caractéristiques il fait le choix le plus convenable à adopter. Sa simplicité de réalisation, sa distribution au niveau des données nous ont poussé à l'envisager pour notre architecture (PADMA-RAD).

Conclusion :

Dans le premier chapitre, on a abouti au choix d'une plateforme de travail présentant une combinaison entre le Data Mining et les systèmes multi-agents, donc l'architecture PADMA servira de base pour concevoir notre travail.

Dans le but d'améliorer le système envisagé, on va apporter certains changements qui consisteront à remplacer la méthode du clustering par l'un des algorithmes étudiés dans le chapitre présent et qui sont destinés à l'extraction des règles d'association distribuées.

On a choisi l'algorithme « Intelligent Data Distribution » pour ses multiples avantages. Il règle le problème de réduction des accès aux sources de données, en plus il convient le mieux aux descriptions du système qu'on veut élaborer.

Dans le chapitre suivant, on dévoilera les phases nécessaires à la conception théorique de notre projet qu'on va intituler « PADMA-RAD » et décrire son principe de fonctionnement tout comme on montrera les mises à jours et les nouvelles adaptations mises en places.

Démarche Méthodologique

-Introduction

-Algorithme IDD

-Architecture PADMA-RAD

-Schéma conceptuel théorique global

-Conclusion

Chapitre 03 : DEMARCHE METHODOLOGIQUE

3.1-Introduction :

Après avoir désigné le modèle d'architecture (PADMA-RAD) qui servira de plateforme de base pour notre travail, on a ensuite étudié les différentes méthodes appliquées au Data Mining distribué basé sur les agents. On a abouti à un choix, celui de l'intégration d'un algorithme à l'approche APRIORI utilisé pour l'extraction des règles d'association distribuées, c'est l'IDD (Intelligent Data Distribution Algorithm).

Ce chapitre est divisé en deux parties, on va d'abord présenter l'algorithme IDD dédié à l'extraction des règles d'association en mode distribué d'une manière plus détaillée. On va aussi donner un exemple pour démontrer son fonctionnement et ses performances.

Après cela, on va passer à l'étape de la conception théorique globale de notre système, qui s'intitule PADMA-RAD signifie « **PA**rrallel **D**ata **M**ining based **A**gent » et RAD signifie « Règles d'Association Distribuées » pour préciser que notre système est basé sur l'extraction des règles d'association distribuées, ce qui diffère au système d'origine qui est basé sur le « Clustering ».

3.2-L'Algorithme IDD :

3.2.1-La Contribution de l'Algorithme IDD :

Comme déjà décrit, IDD vient résoudre certains problèmes de l'exécution de l'algorithme DD. Toutefois le principe reste simple, les partitions de données stockées dans des bases de données sont envoyées à travers les autres sites grâce à un réseau virtuel en anneaux basé sur le passage de le passage de jeton entre tous les sites (all to all broad cast) [16,13W, 02M].

Cette structure d'anneau ne pose pas de problème en elle-même, de plus elle est réalisable dans toutes les architectures distribuées. l'algorithme (pseudo-code) permettant de réaliser un tel anneau est présenté dans la figure ci-dessous, sachant que chaque processus possède un Buffer pour la réception des données (RBuf) , un Buffer pour l'envoie des données (SBuff).

```
While (!done) {
    FillBuffer(fd,SBuf);
    For (k=0; k< P-1 ; ++k ){
     /* envoyer&recevoir les données dans
     un pipeline non bloqué */
     MPI.Irecv(RBuf , left);
     MPI.ISend(SBuf , right);
     /* traiter la transaction dans SBuf et
       mettre à jours l'arbre de hashage HTree */
     Subset(HTree , SBuf);
     MPI.Waittall();

     /* interchanger les buffers */
     tmp=SBuf;
     SBuf=RBuf;
     RBuf=tmp;
    }
/*traiter la transaction dans SBuf et mettre à
jours l'arbre de hashage  HTree   */
Subset(Htree,SBuf);
}
```

Figure9 : Pseudo-code pour le « Mouvement des données»

Initialement , un Sbuff est chargé de données locales, chaque processus commence de manière asynchrone à envoyer ces données à son voisin (Droite) tout en recevant depuis l'autre voisin (Gauche) un paquet qu'il range dans le Rbuff.

Quand tous les processus auront appliqué l'opération de manière asynchrone, chaque processus va traiter les informations qu'il a reçues et obtient à partir de cette transaction une collection d'items. La fin de l'opération désignera la liste des candidats (itemset) pour chaque processeur. Ce qui diffère à l'algorithme DD, La communication n'est possible qu'entre chaque processeur et son voisin gauche ou droite.

Pour éliminer les redondances dans les itemset candidats, une procédure de partitionnement est prévue par chaque processeur et une comparaison de ses données avec ses voisins les plus proches. Ce type de transaction des sous-ensembles détermine si les partitions des voisins contiennent l'item en question en vérifiant l'arbre de hachage (le découpage des données).

3.2.2-Principe du Partitionnement Intelligent :

On a quatre processeurs P0,P1,P2,P3. Ils disposent de deux supports de stockage

Chaque processeur exécute le partitionnement de manière asynchrone.

Soit un ensemble d'items pour proc0 L={1,2,3,4,5,6,7,8}.

Le principe du partitionnement consiste à ce que chaque processus garde un ensemble d'items et fait passer les items-set non traités aux autres processus , exemple : pendant que P0 s'occupe de 1 et 7, P1 traite 2 et 5, P2 traite 4 [02M, 16]

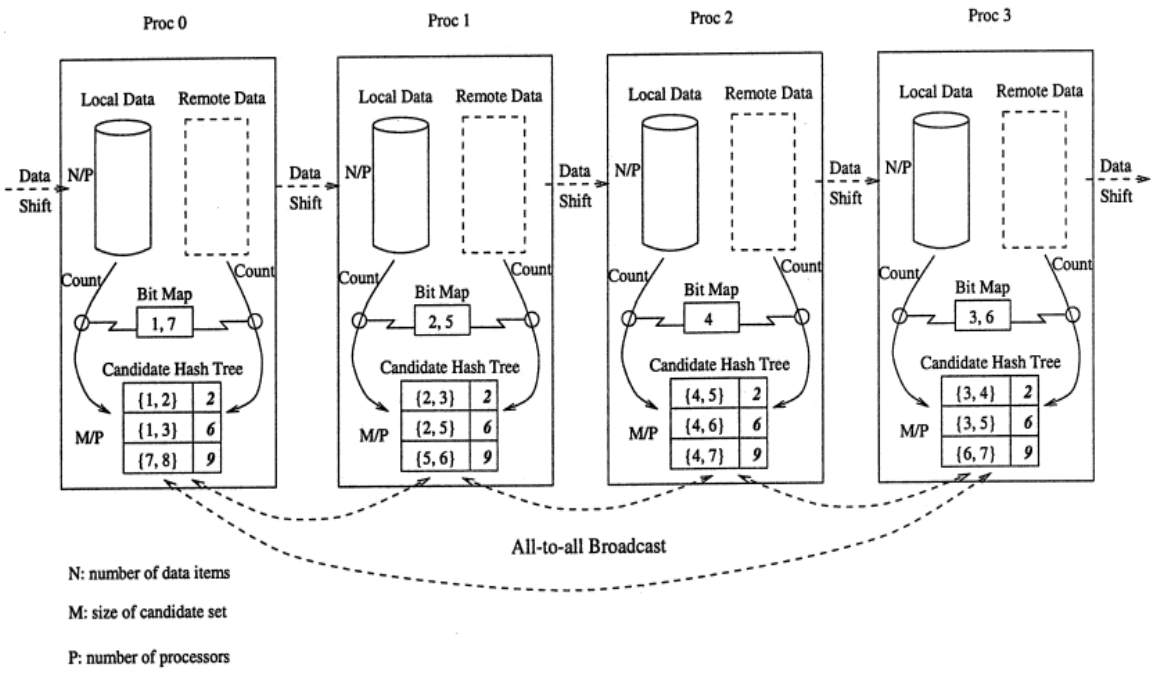


Figure10 : Algorithme IDD « Intelligent Data Distribution »

L'étape la plus importante qui donne à l'algorithme IDD sa spécifité est celle du partitionnement intelligent des données distribuées, alors que le reste des étapes de l'Algorithme IDD s'applique de la même manière que le DD,

La section prochaine montre le rôle de l'algorithme dans un système Data Mining distribué basé sur les agents, et à quel niveau est son implantation.

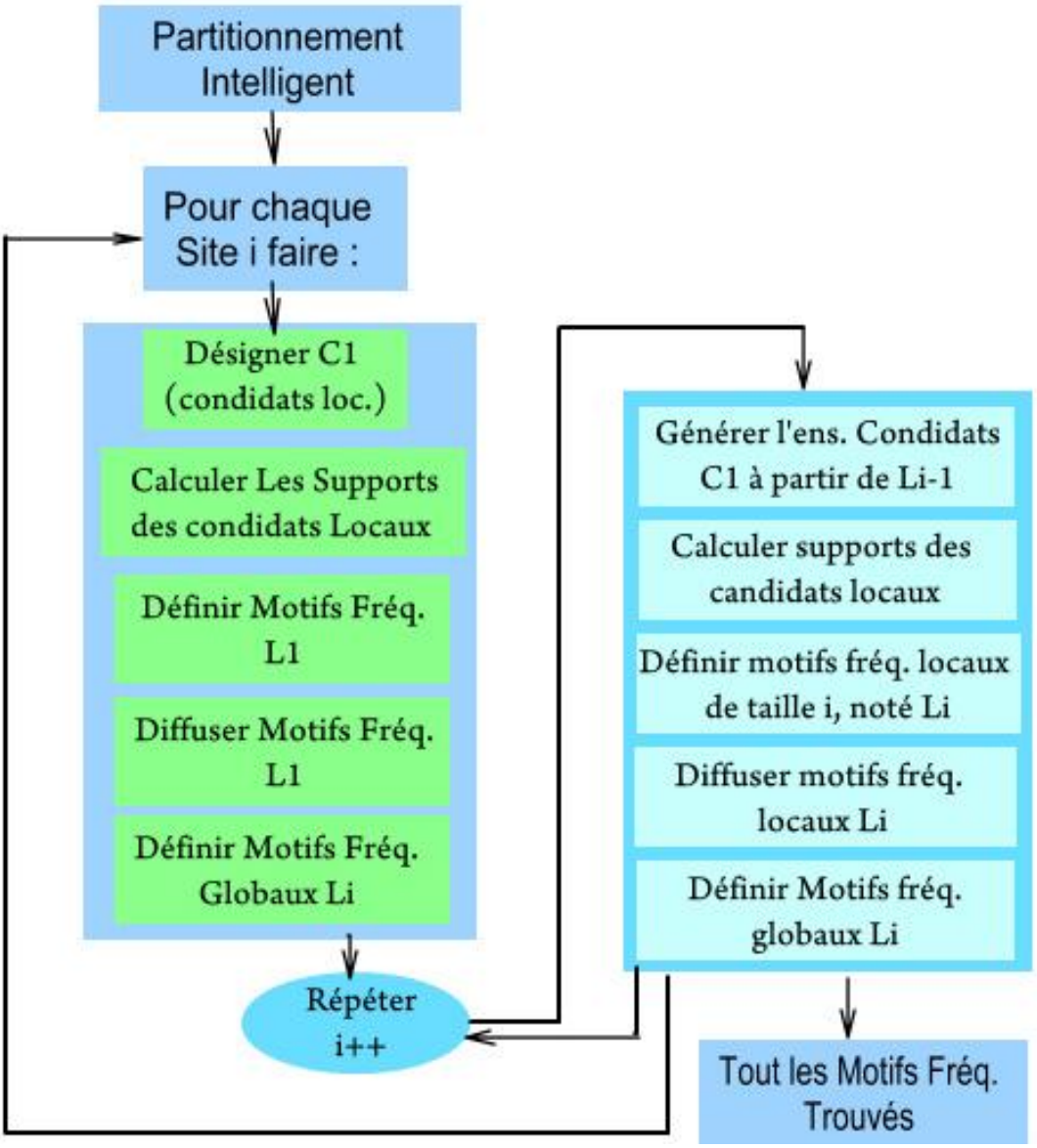


Figure11 : Algorithme IDD « Intelligent Data Distribution »

Exemple d'application de l'algorithme Intelligent Data Distribution (Extraction des Règles) :

Student Choice Data		
Roll No	Student Choice_ Sequence	Branch_ ID
1001	1	4
1002	2	2
1003	3	6
1004	4	11
1005	5	7
1006	6	9

Tableau 2 : Les données concernant les choix d'admission d'un groupe d'étudiants.

Branch ID Generation		
Branch_ ID	College_Name	Branch
1	C1	CS
2	C1	IT
3	C1	EC
4	C2	CS
5	C2	IT

Tableau 3 : les Branches (BRANCH_ID) et les Collège qui correspondent

Transactional Data Set				
Transaction_ID	A	B	C	D
T1	1	1	1	1
T2	1	0	1	1
T3	1	1	0	1
T4	1	0	1	1
T5	1	0	1	1
T6	1	0	1	1

Tableau 4 : les transactions issues du Tableau2

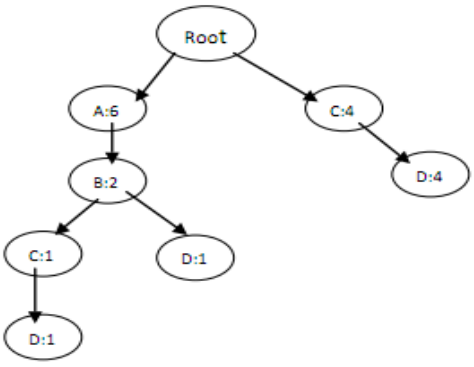


Figure 12 : L'arbre de déduction des Règles d'Associations

Rule	Confidence
C1 -> C3	83.33333
C3 -> C1	100
C1 -> C4	100
C4 -> C1	100
C3 -> C4	100
C4 -> C3	83.33333
C1, C3 -> C4	100
C4 -> C1, C3	83.33333
C1 -> C3 ,C4	83.33333
C3, C4 -> C1	100

Tableau 5 : La Liste des Règles d'association

Remarque : support minimum = 70%, le taux minimal de confiance est 80 %

A partir des tableaux 2 et 3, l'algorithme (implémenté dans un outil logiciel) recrée les données sous forme disjonctive binaire (Tableau 4), ensuite il extrait les règles d'association sous forme d'approche de ressemblance (Confiance) ; le tableau 5 regroupe ces règles suivant un niveau minimal de ressemblance.

La section qui va suivre va expliquer à quel niveau dans l'architecture du système l'algorithme d'extraction des Règles d'associations est implanté, on va donner un schéma conceptuel globale et détaillé sur le système à mettre en œuvre, enfin on vé définir ses différentes composantes.

3.3-L'Architecture PADMA-RAD :

3.3.1-Présentation :

Le but d'avoir choisi l'algorithme IDD est basé sur le gain de temps d'accès aux sources de données, cela pour alléger notre approche de l'extraction des règles d'association qui va servir de moyen d'optimiser le système de fouille de données basé sur les agents.

Comme décrit dans le chapitre précédent, l'architecture parallèle baptisée PADMA est souvent conçue autour des agents Data Mining, qui fonctionnent grâce à l'algorithme du clustering. A la différence de ces projets, dans notre cas, les agents Data Mining sont montés sur l'algorithme d'extraction des règles d'association distribuées, adoptant l'approche APRIORI.

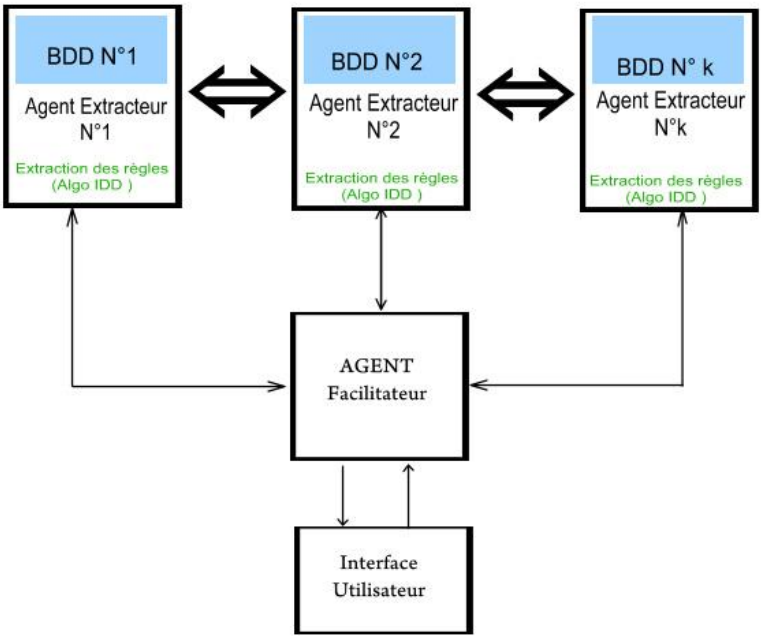


Figure 13 : Schéma Conceptuel Général du PADMA – RAD

3.3.2-Distribution des Tâches :

Dans le cas d'une architecture comme PADMA-RAD, les tâches du processus de fouille de données sont toutes confiées aux agents Data Mining (Collecte d'infos, prétraitement…etc.).

L'extraction des connaissances se fera par le billet d'un algorithme dit APRIORI qui a le but de l'extraction des règles d'association entre les différents itemset trouvé dans les bases de données.

Le système sera constitué spécifiquement de trois grands niveaux, le premier est celui des Agents de fouille, le second est constitué d'un agent facilitateur, à la fin c'est le niveau de l'interface utilisateur. On va expliquer les tâches effectue à chaque niveau [11W, 18].

a)-Les agents de fouille :

Comme il a été expliqué précédemment, les agents Data Mining sont des programmes autonomes qui se chargeront de collecter, prétraiter les informations et d'en extraire les connaissances utiles, ils respectent l'algorithme choisi par leur concepteur. Ils sont sensés rapporter leurs résultats respectifs à l'agent facilitateur.

b)- L'agent Facilitateur :

Pour chaque nombre d'agents Data Mining, on désigne un agent facilitateur ou agent central, il facilite la communication entre chaque agent et un autre , ou entre les agents et l'utilisateur, c'est lui qui interprète la requête de l'utilisateur, distribue les opérations et interprète le résultat en retour.

c)- L'interface utilisateur :

L'interface permet à l'utilisateur d'exprimer sa demande (requête) et l'envoie à l'agent facilitateur, puis récupère les résultats ce de dernier pour les afficher .

3.4-Schéma de Conception théorique global :

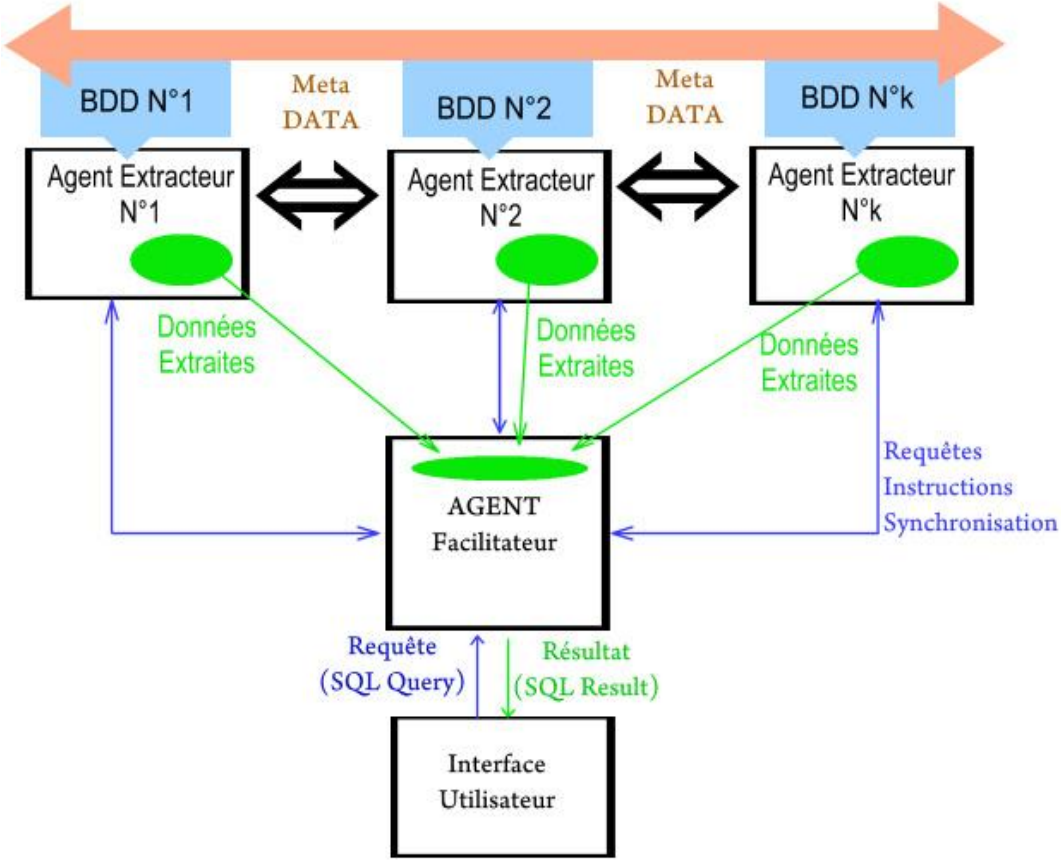


Figure 14 : Schéma Conceptuel Détaillé du PADMA – RAD

Principe de Fonctionnement PADMA-RAD :

Comme décrit dans le schéma ci-dessus, notre nouveau système (PADMA-RAD) fonctionne suivant les étapes suivantes [11W, 13W] :

a) Initialisation :

D'abord l'utilisateur propose une requête pour consulter des données contenues dans une série de datasets à travers un réseau réparti, c'est l'interface utilisateur qui prend en charge cette requête. C'est là où la structure du système se met en place, un premier agent central (agent facilitateur) est mis à disposition pour interpréter la demande de l'utilisateur et définit les paramètres de lancement du processus de fouille (nombre de sites à visiter, configuration des agents de fouille…etc.) voir figure 15.

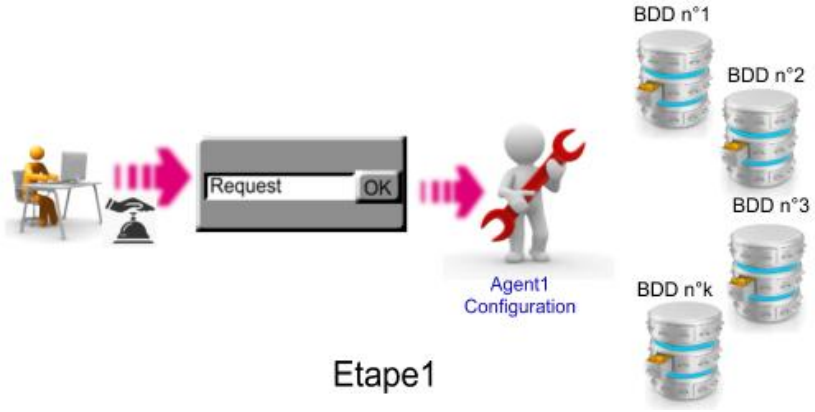


Figure 15 : Représentation de l'étape d'initialisation

b) Partitionnement :

Après l'installation des agents d'exploration et la finalisation de l'étape d'initialisation du système, la première tâche se déroule au niveau des agents Mining avant le lancement de la fouille, c'est l'étape qu'on va appeler « Meta-Data » (voir la figure 14 et 16, couleur orange) où les agents avant de commencer leur travail vont se mettre d'accord sur les paramètres de partitionnement des données locales et globales entre eux, cela va les aider à gagner du temps dans leurs transactions.

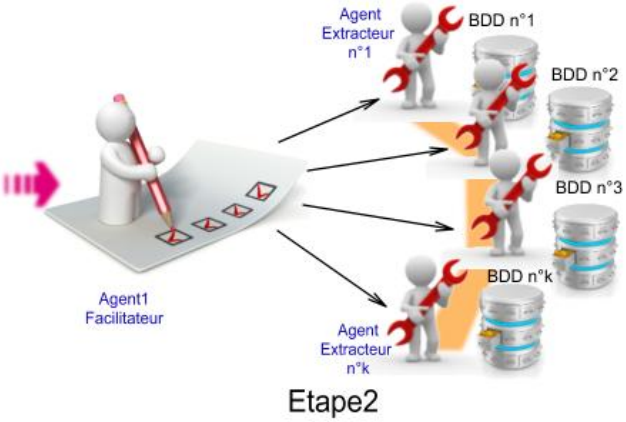


Figure 16 : Représentation de l'étape Meta Data

c) Fouille de Données :

Chaque agent doit exécuter indépendamment la requête et lance sa propre fouille en interrogeant l'ensemble de données (dataset) qui lui est associé. Cette indépendance dans la phase initiale permet d'accélérer le processus Data Mining (voir la figure 17, couleur bleu).

De temps en temps, les agents Data Mining doivent communiquer, pour s'échanger des informations concernant les partitions de données à traiter, c'est-à-dire repasser par l'étape « Meta-Data », parfois ces informations peuvent concerner aussi d'autres interactions (voir la figure 17 couleur verte).

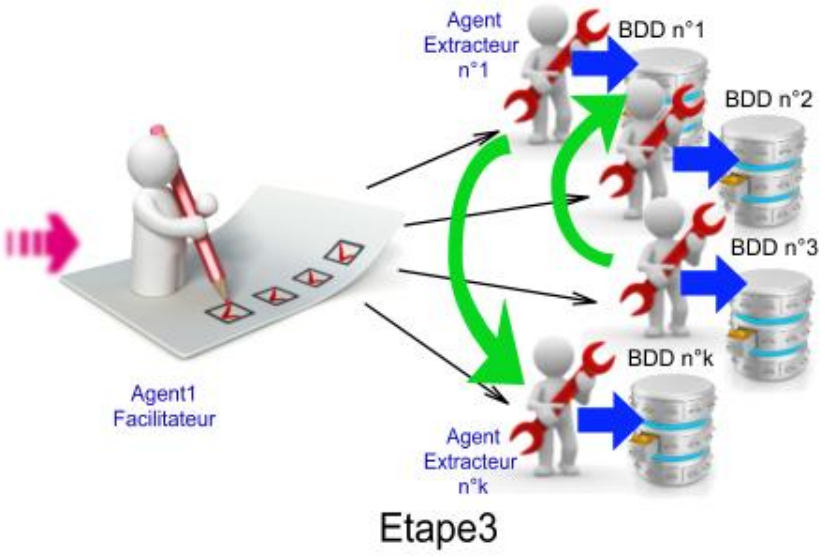


Figure 17 : Représentation de l'étape Data Mining

Après l'extraction de données par les agents, le facilitateur les intègre dans un ensemble de données global. PADMA analysent les données dans une Manière parallèle. Le facilitateur ordonne les agents Data Mining pour exécuter un algorithme d'extraction des règles (RAD- IDD) sur leurs sources de données locales respectives et leur voisinage. Chaque agent communique un ensemble de motifs au facilitateur sans interagir avec les autres agents. Le facilitateur combine alors les motifs des agents et renvoie le résultat à l'interface utilisateur.

Figure 18 : Représentation de l'étape Interprétation du Résultat

Dans la prochaine rubrique, on va éclaircir un point très important pour la mise en œuvre de notre système, c'est l'architecture théorique des différents agents exploités dans notre travail.

d) Structure conceptuelle des Agents :

-Agent Central (le Facilitateur) :

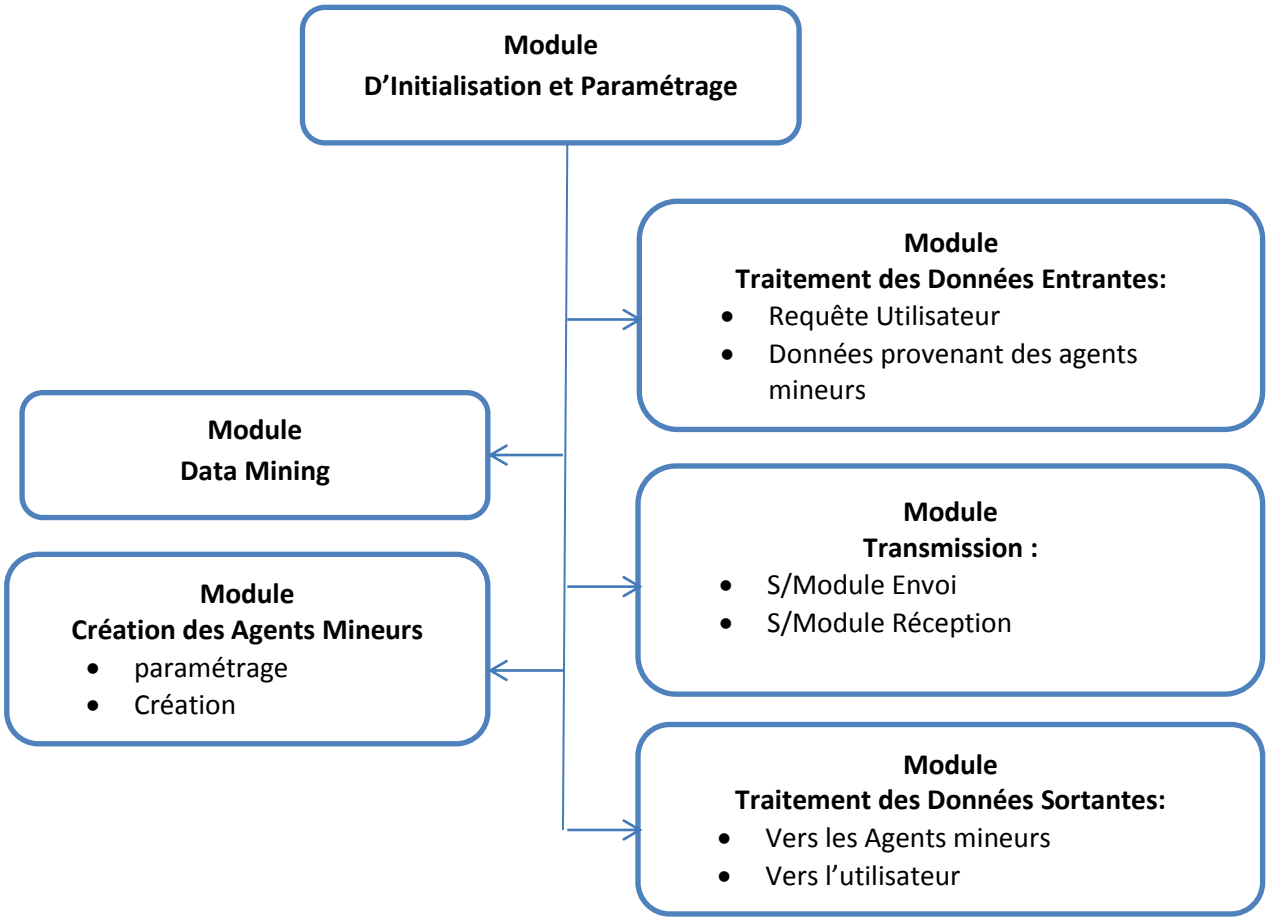


Figure 19 : Structure conceptuelle de l'agent central (facilitateur)

Comme le montre la figure 21, l'agent facilitateur est composé de six modules principaux, chacun des modules prend en charge l'exécution de certaines tâches, chacun d'eux peut à son tour être constitué de sous modules.

Le module d'initialisation est le point central du fonctionnement de l'agent central ou l'agent facilitateur, car c'est à partir des données de ce module que dépend le reste. Le module de création des agents mineurs est le module qui caractérise l'agent central (le facilitateur) et qui l'aide à lancer les processus data mining en mode distribué.

Quand on n'a pas affaire à un système distribué (mode mono source), le module data mining permet à l'agent central de jouer le rôle d'un simple agent mineur, sinon il s'en sert pour traiter les données prévenantes des autres agents.

-Agent Data Mining (Agents Mineurs) :

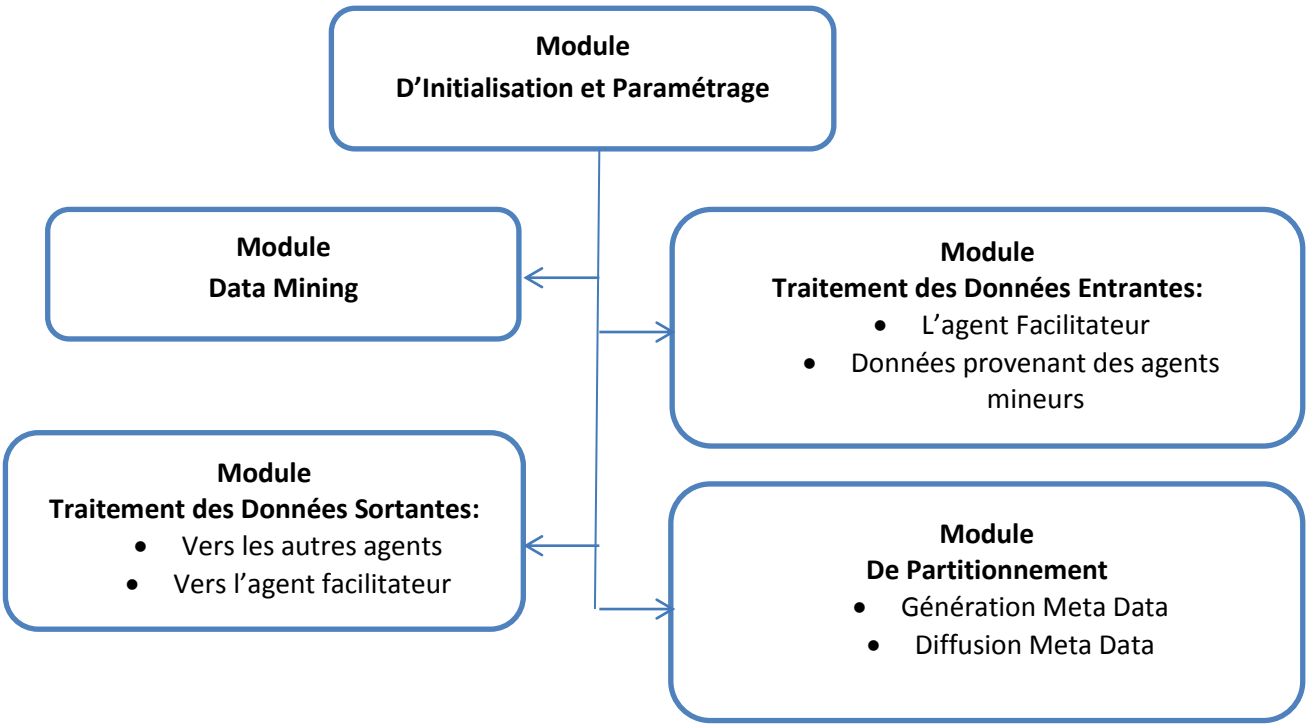


Figure 20 : Structure conceptuelle de l'agent mineur

L'agent mineur est conçu de la même manière que le facilitateur. Tous les modules qui le composent fonctionnent autour d'un module central, c'est le module d'initialisation et de paramétrage. C'est lui qui fournit les données support sur lesquelles le travail local et global va se baser.

La seule différence dans sa structure est qu'il ne contient pas de module de création, donc il se contente de partitionner les données locales et de lancer un processus de fouille à son niveau tout en communiquant avec ses voisins.

3.5-Conclusion :

Ce chapitre compte parmi les plus importants de ce mémoire, car il explique théoriquement le travail demandé dans sa totalité. Après cela on a bien une vision globale sur le système à mettre en œuvre et c'est avec ces concepts qu'on va apporter nos modifications.

Le prochain chapitre décrit les démarches techniques de la mise en œuvre du projet PADMA-RAD et le test de ses performances tout en montrant les points forts résultant de sa personnalisation.

Implémentation et Test Opérationnel

-Introduction

-Implémentation

-Teste Opérationnel

-Conclusion

Chapitre4 : Implémentation et test Opérationnel

I. Introduction :

Ce chapitre décrit l'étape finale concernant la mise en œuvre du système PADMA-RAD, il explique la phase de l'implémentation, le déploiement et le test des performances de ce dernier.

Cette dernière phase de notre travail est composée elle-même en trois démarches importantes. En premier lieu, on commence par le processus d'implémentation des différentes parties du projet, c'est la mise en œuvre physique du projet PADMA-RAD qui est le fruit d'une multitude d'études et de préparations décrites à travers les chapitres précédents. Ensuite, on définit un environnement d'essai pour le projet, cela signifie qu'on lui trouve un domaine d'application, sur cette idée on va bâtir une base de données représentant des données réelles ou semi-réelle et son propre environnement d'exécution.

En dernier, le test opérationnel sera la troisième démarche et la finale, on va mettre le projet final qui résulte du processus d'implémentation dans un environnement regroupant les bons paramètres de fonctionnement. Le procédé de test et d'évaluation doit pourvoir s'appliquer sur le projet « PADMA-RAD » selon deux approches dont la première vient en se servant de la base de données et du contexte d'application proposés par le développeur, la deuxième approche consiste à sélectionner une base de données synthétique obtenue à partir d'autres sites (sur internet).

II. Implémentation :

II.1 Environnement et Plateforme de travail :

Pour mettre en œuvre un tel projet, plusieurs environnements de développement dits modernes ont été mis à la disposition des développeurs, aux chercheurs et aux innovateurs afin de pouvoir concevoir des projets pouvant atteindre les plus hauts sommets de la complexité : .Net, JAVA, C #.

Comme tout concepteur, le chercheur doit prendre en compte certains paramètres afin de garantir le bon aboutissement de son travail tel que : la modularité, l'aspect orienté objet, le multiplateforme, rapidité d'exécution et fiabilité.

L'environnement de travail choisis pour notre projet est décrit dans le tableau suivant :

Type de la plateforme	Nom de plateforme	Description
Equipement matériel	Ordinateurs	Intel – dual core 1 et 2 Go RAM 160 et 500 Go HDD Connectés à internet
Système d'exploitation	Windows XP (SP 3) , Windows 7	/
Environnement de d'exécution	Plateforme JEE	Java Enterprise Edition
Plateforme Multi-Agents	JADE	Implémenté en java
Environnement utilisateur pour le développement	Eclipse	IDE pour java
Plateforme de la Base de Données	Environnement MySQL	Outil utilisé : easyPHP

Tableau6 : L'environnement logiciel et matériel de mise en œuvre de PADMA-RAD

Présentation de la plateforme JADE :

Jade est un middleware qui facilite le développement des systèmes multi agents (SMA). JADE contient :

- **Un runtime Environment :** l'environnement ou les agents peuvent vivre. Ce runtime environnement doit être activé pour pouvoir lancer les agents.
- **Une librairie de classes :** que les développeurs utilisent pour écrire leurs agents
- **Une suite d'outils graphiques :** qui facilitent la gestion et la supervision de la plateforme des agents

Chaque instance du JADE est appelée conteneur " container ", et peut contenir plusieurs agents. Un ensemble de conteneurs constituent une plateforme. Chaque plateforme doit contenir un conteneur spécial appelé main-container et tous les autres conteneurs s'enregistrent auprès de celui-là dès leur lancement [15,14W,18W].

La figure suivante illustre les concepts de base du jade en montrant un petit exemple de deux plateformes jade composées respectivement de trois et un conteneur Chaque agent est identifié par un identifiant unique et peut communiquer avec n'importe quel autre agent sans avoir besoin de connaître son emplacement :

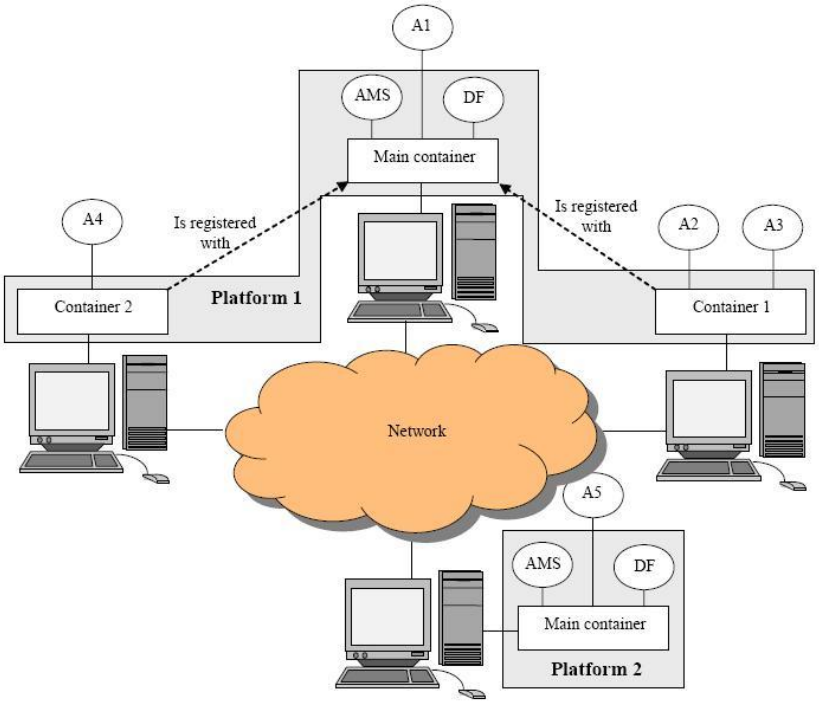


Figure 21 : la répartition de la plateforme JADE en conteneurs

Pour bien comprendre le principe de mise en place et le fonctionnement de la plateforme multi agents JADE, le développeur doit retenir les concepts suivant [15,14W,18W] :

L'installation et la configuration de la plateforme, la notion de conteneurs et la création des agents, tout cela est expliqué dans l'annexe 01 (A1,A2,A3,A4).

II.2 Mise en Œuvre du projet PADMA-RAD :

Présentation :

Cette section est une brève explication du parcours des différentes phases d'implémentation de notre plateforme « PADMA-RAD ». Elle comprend à son tour plusieurs parties expliquant chacune le côté de la réalisation de notre projet.

Les différentes parties de cette section seront décrite comme suit : On va d'abord donner une explication sur la structure logique du système envisagé, ensuite on va dévoiler le schéma structurel technique qui va représenter la globalité du système du point de vue fonctionnel et modulaire ; Après cela, on va donner quelques extraits du code d'implémentation du projet pour mieux expliquer sa phase de réalisation où on va citer les différentes parties du programme et les paramétrages nécessaires à sa mise en route.

La partie d'implémentation ne peut être décrite en sa totalité pour sa complexité et son volume, pour des raisons de simplicité et limitation de temps. On s'est contenté de ne décrire que les parties les plus essentielles.

Structure Logique du Système :

Afin de maintenir le sens global du sujet étudié, l'étude de la contribution des agents dans un processus de Fouille de données basées sur l'algorithme APRIORI, on devait donner au système mis en œuvre une structure cohérente, souple et interactive pour permettre de donner la chance à l'utilisateur de lancer un processus d'extraction de données et de choisir le mode de Fouille (avec agents ou sans agents).

Le système envisagé doit donner la main à l'utilisateur de résoudre la même problématique selon trois méthodes différentes, ce qui donne la possibilité de comparer entre les trois différentes méthodes de résolution d'où le but global du projet. Chacune de ces trois options dites méthodes vont représenter respectivement des processus de fouille de données distincts, ce qui va être expliqué dans le prochain paragraphe.

C'est pour cette raison que notre système sera bâti sur trois niveaux différents. Le premier décrivant un processus d'extraction basé sur l'algorithme APRIORI mais en mode non distribué, pas d'utilisation des agents ; Le deuxième niveau décrit un processus de fouille aussi basé APRIORI et en mode distribué ; Le troisième désigne un processus d'extraction exécuté par des agents mais cette fois en suivant l'algorithme IDD.

Ainsi, avec une telle architecture on aura laissé la possibilité à l'utilisateur de comparer les trois types de processus Data Mining du point de vue vitesse de rendement et précision du résultat.

Le système décrit aura une architecture comme celle décrite dans la rubrique suivante, (voir figure 22).

Schéma Structurel Technique du Système PADMA-RAD :

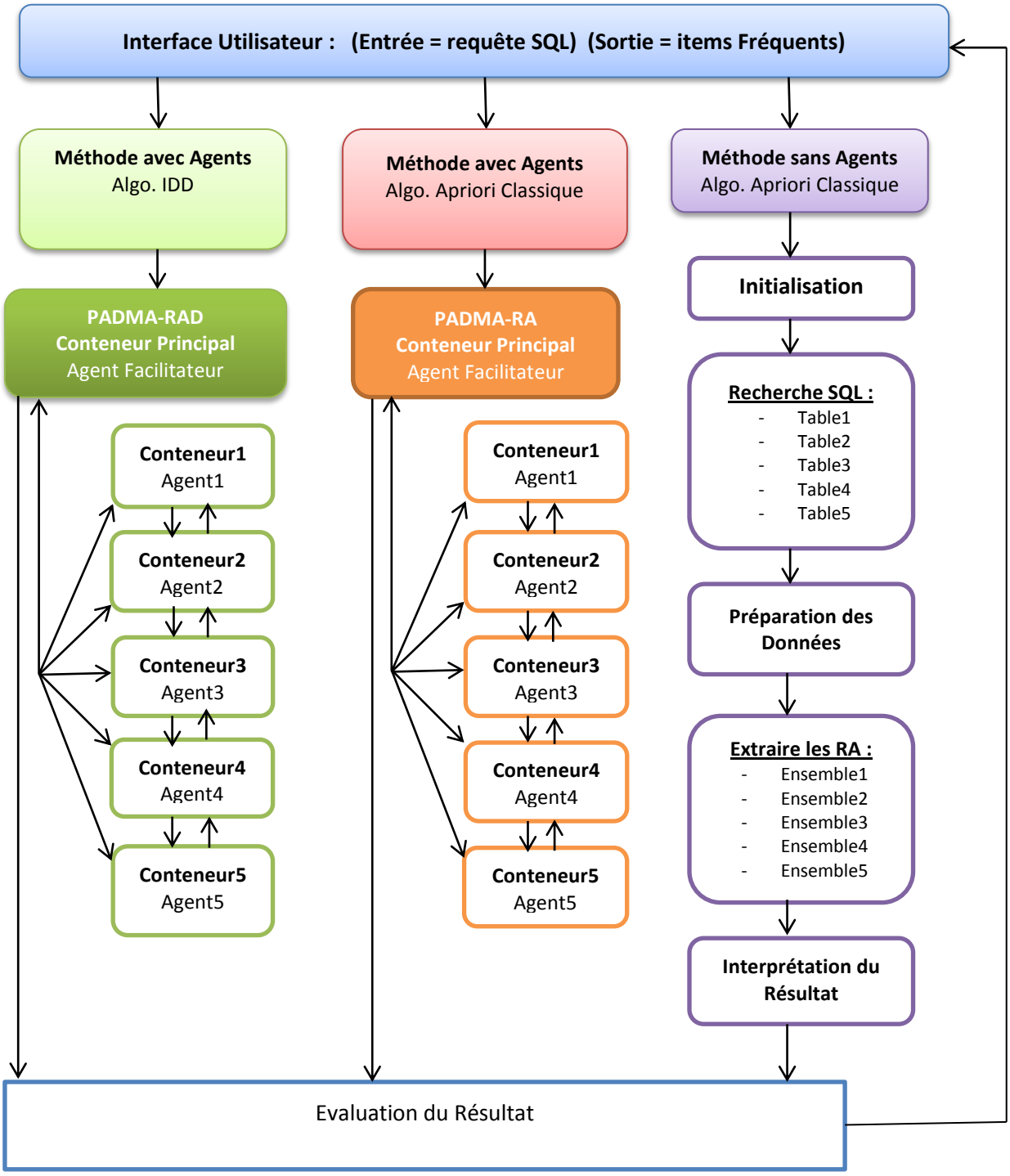


Figure 22 : Architecture PADMA-RAD

Remarque : Comme il est montré dans la figure 22, le système contient deux répliques de la structure PADMA, la première inclut l'utilisation de l'algorithme APRIORI classique, la seconde inclut l'utilisation de l'algorithme IDD.

Structure modulaire de l'Agent Central (Le Facilitateur) :

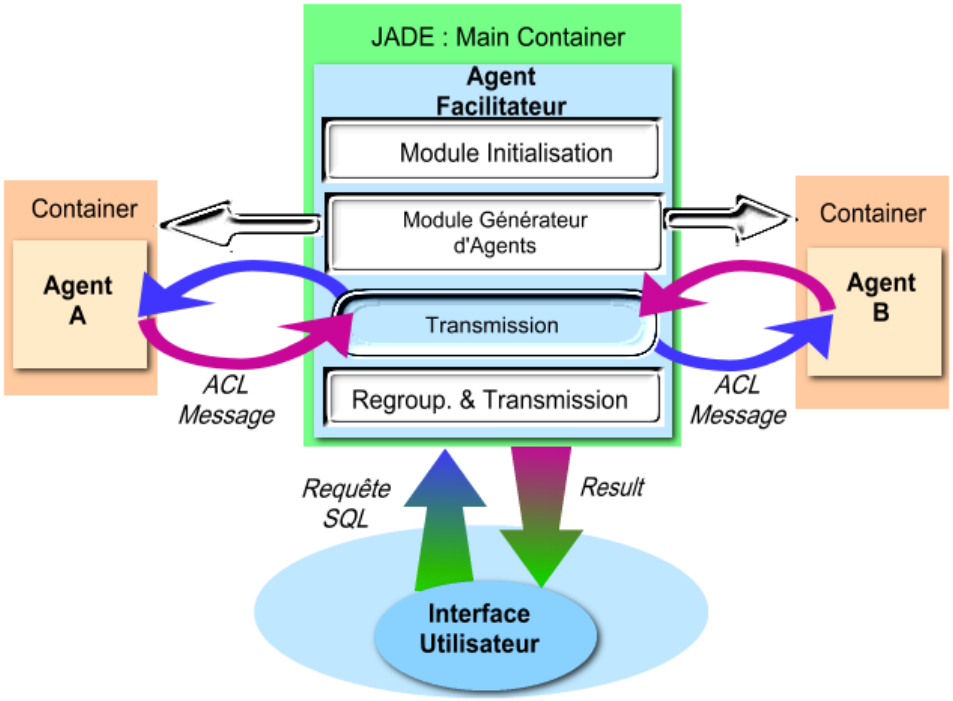


Figure 23 : Architecture modulaire d'un agent facilitateur sous JADE

Comme montré dans la figure 23, l'agent facilitateur est déployé dans son propre conteneur JADE. Ce conteneur est considéré comme le conteneur principal de notre système. L'agent facilitateur communique de manière indépendante avec l'interface utilisateur, depuis son conteneur il reçoit la requête de l'utilisateur sous forme d'une requête SQL et la description du mode de fouille de données , ensuite il est sensé retourner un résultats sous forme textuelle contenant les ensembles d'items fréquents regroupé et la durée du traitement.

L'agent Facilitateur est composé en plusieurs blocs ou modules, le premier d'entre eux est le module d'initialisation qui permet à ce type d'agent de récupérer ses paramètres de fonctionnement.

Grâce au module d'initialisation, le facilitateur peut aussi récupérer les données permettant de lancer son deuxième bloc intitulé « bloc de génération des agents », c'est grâce à ce dernier que notre facilitateur peut déployer les agents extracteur dans leurs conteneurs respectifs et les initialiser.

Il reste deux autres blocs, un pour permettre au facilitateur de communiquer avec les agents extracteurs par l'échange de message de type ACLMessage, le module restant permet de regrouper les données envoyées par les extracteurs et transmettre le résultat à l'interface principale pour être affichée à l'utilisateur.

Structure modulaire de l'Agent Extracteur (Data Mining Agent) :

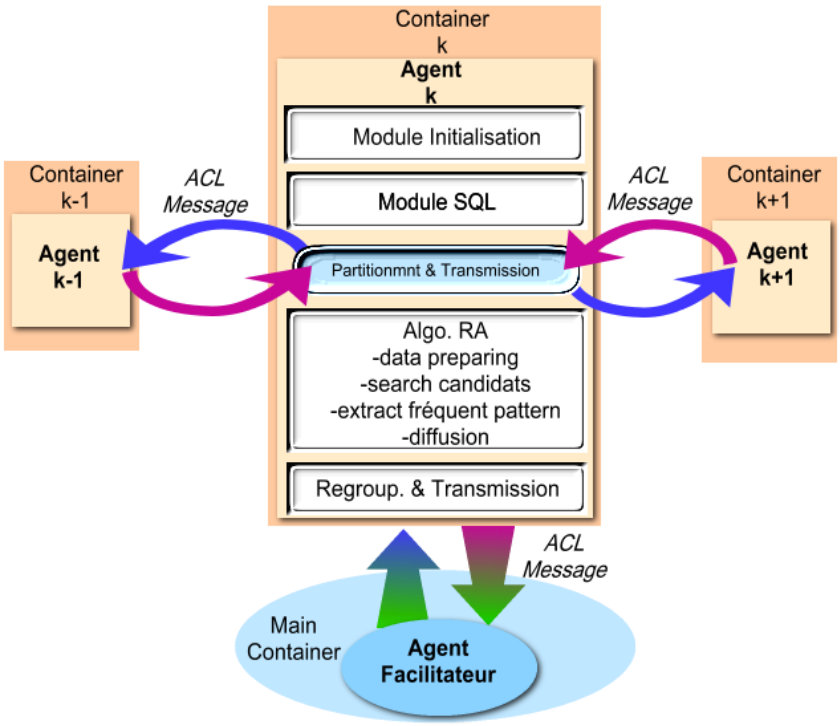


Figure 24 : Architecture modulaire d'un agent Extracteur sous JADE

Chaque agent extracteur est conçu autour d'une structure composée de 5 modules (voir figure 24)

- **Module d'initialisation :** il permet de regrouper les données indispensable au fonctionnement de l'agent et au processus de fouille qu'il va exécuter.
- **Module SQL :** chaque agent doit récupérer la requête SQL envoyée par l'utilisateur pet passant par le facilitateur, une recherche SQL est ensuite appliquée pour générer un ensemble de données initial sur lequel le travail pourra débuter.
- **Module partitionnement& transmission :** composé à son tour de deux sous blocs

Bloc transmission : permet aux agent extracteur de s'échanger les données résultant du Module SQL afin de paramétrer le module de recherche des Règles d'association.

- **Bloc partitionnement :** il est activé si l'algorithme IDD est choisis par l'utilisateur, car il prend en charge la phase de partitionnement des données qui précède l'extraction des règles d'association en mode distribué (cas IDD).

La rubrique qui va suivre donnera un léger aperçu sur l'implémentation de notre système en sa globalité, pour des raisons de légèreté du contenu, on ne citera que les points importants du code source .

Diagramme des Classes (représentation UML2.0) :

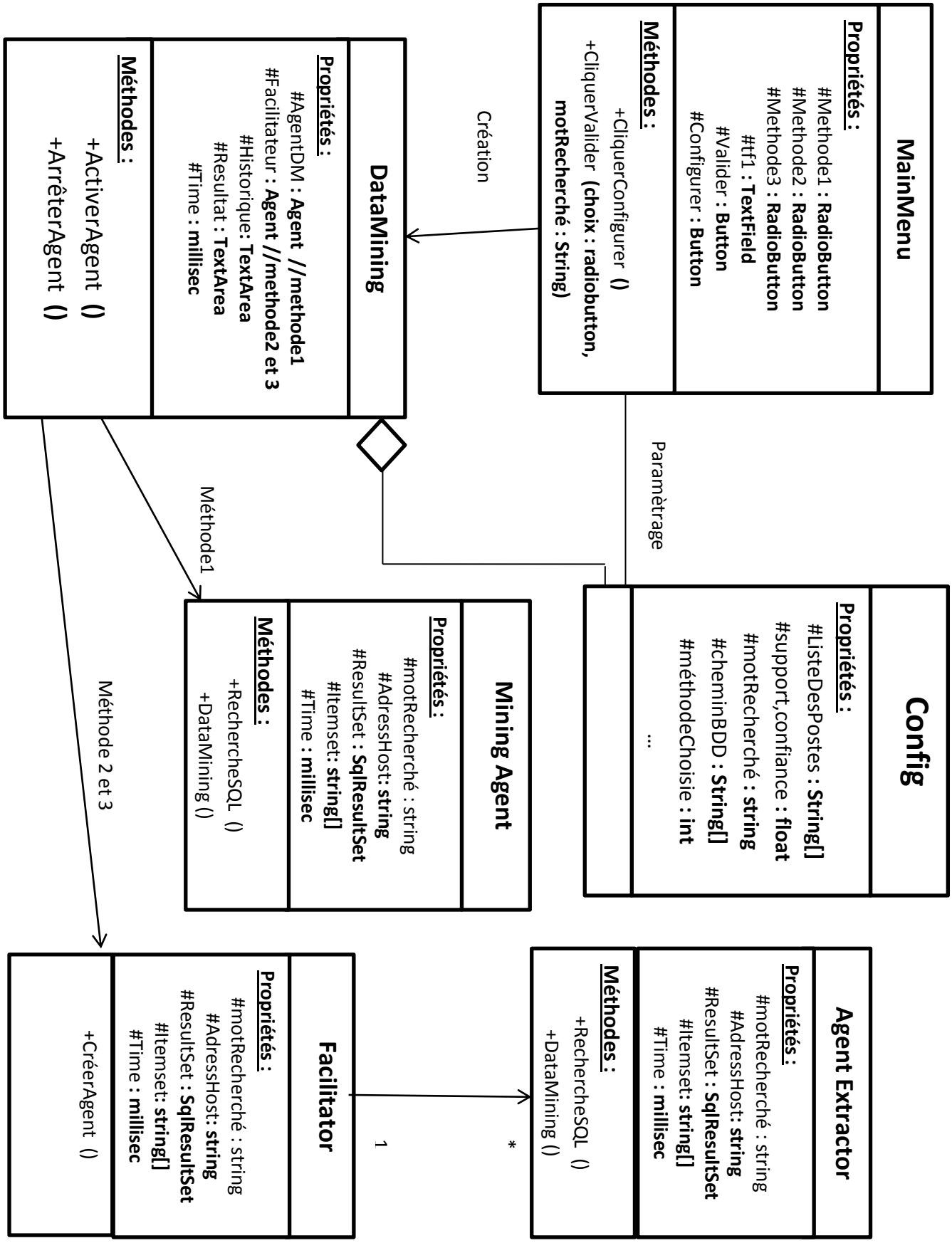


Figure 25 : Diagramme de Classes (UML2.0) de PADMA-RAD

Quelques Extraits de la phase d'Implémentation :

Le processus d'implémentation du projet est centré sur deux concepts clés, l'implémentation de la plateforme JADE (création de conteneurs, création d'agents, gestion des comportements « behaviour ») et l'implémentation des classes de type Thread en java (création de thread, lancement de processus, communication et renvoie de paramètres).

L'implémentation du système sur la plateforme JADE comprend à son tour l'appel aux concepts suivants :

- La création du conteneur principale (voir annexe01 A5)
- La création des conteneurs secondaires (voir annexe01 A5)
- Les comportements « behaviours » (voir annexe01 A6)
- La Communication entre Agents (voir annex01e A7)

De l'autre côté, le code d'implémentation des processus java (thread) regroupe les éléments suivants :

- La création d'une classe thread et son lancement (voir annexe01 A8)
- Le code du module de préparation des données pour la fouille (voir annexe01 A9)
- Le code du module de connectivité mySQL (voir annexe01 A10)
- Le code du processus d'extraction de données (règles d'association) (voir annexe01 A11)

III. Domaine d'expérimentation du projet PADMA-RAD :

III.1. Présentation :

La phase d'implémentation du projet PADMARAD s'est déroulé avec succès, pour tester le système et ses performances, il fallait lui créer un contexte d'exécution ou un environnement générant des données qu'il va exploiter. Donc, le système doit après tout avoir un domaine d'application initial pour son expérimentation.

Après avoir créé un prototype du système, il faut le tester dans son contexte d'exécution initial en lui fournissant une base de données dont la structure respecte son contexte et son domaine d'application (domaine d'expérimentation ciblé).

Pour garantir la réutilisabilité et l'extensibilité du projet, on doit aussi le tester dans des contextes différents que celui de son domaine d'application initial, il faut donc lui fournir des ensembles de données souvent appelés « des données synthétiques » ou des bases de données de synthèse », cela donne le concept d'expérimentation non ciblée.

III.2. Base de Données pour l'expérimentation ciblée :

Pour démontrer le fonctionnement de notre système PADMA-RAD, il fallait mettre en place une base de données qui servira de modèle pour les manipulations SQL, c'est pour cela qu'on a eu recours aux SGBD. L'exemple de SGBD qu'on a choisis a été créé avec la plateforme mySQL pour sa facilité d'utilisation et de ses outils fiables et ergonomiques.

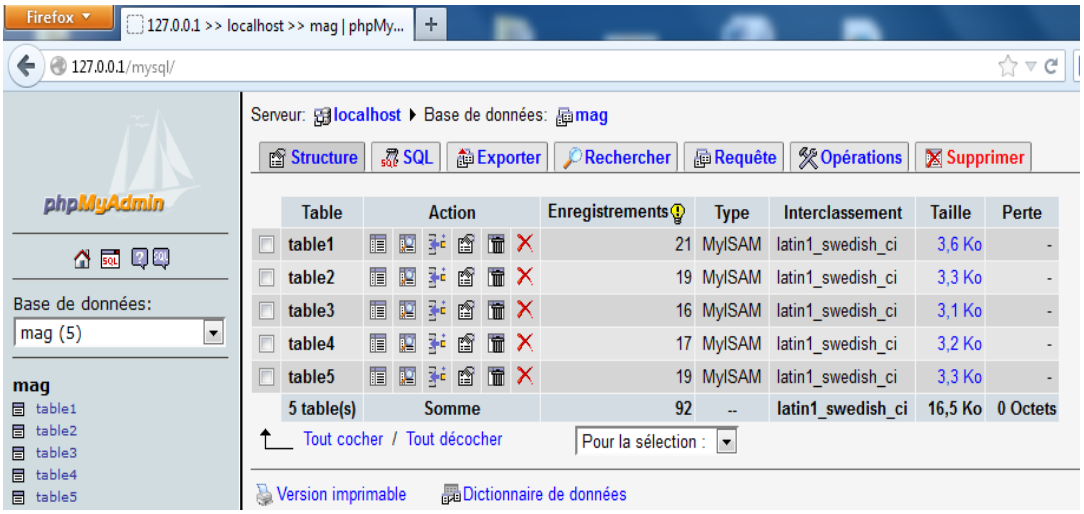


Figure 26 : Aperçu sur l'interface de la plateforme mySQL

On a limité notre surface d'essai à 5 site contenant chacun une table (voir figure 32). La structure des tables sont identiques (homogènes), la structure de la base de données à été définit suivant un modèle servant à regrouper les préférences des internautes pendants leurs recherches.

Quand un internaute se connecte et énonce sa recherche, on note les termes qu'il a utilisé, le résultat apparu (l'adresse de la page visitée) et l'ordinateur qui héberge la page, il y a d'autres détails pris en comptes. Tout cela est enregistré dans notre table modèle (voir figure 33)

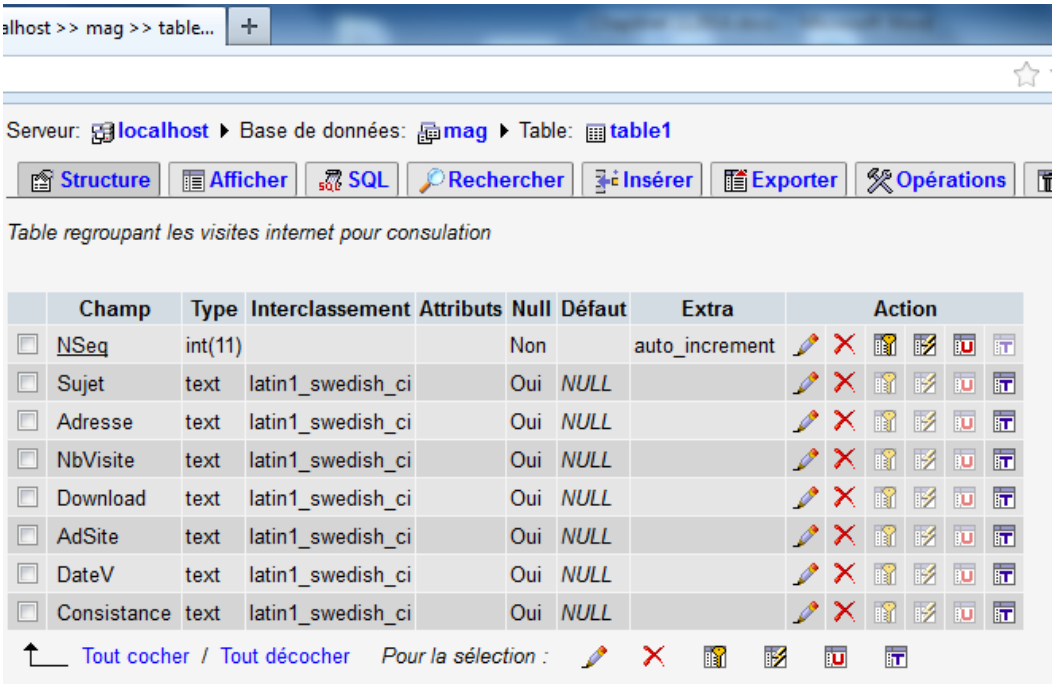


Figure 27 : Aperçu sur la Structure de la Table prise comme modèle

La base de données sélectionnée pour l'expérimentation ciblée n'est rien qu'une table ou l'on sauvegarde les historiques concernant les manipulations quotidiennes sur le Web, ce qui signifie que notre système doit fouiller des données concernant les accès aux différents sites web pour en déduire les préférences des utilisateurs.

III.3. données synthétique pour l'expérimentation non ciblée :

Présentation :

L'un des plus grands avantages de l'architecture PADMA qui a servi de base pour notre projet c'est l'extensibilité et la réutilisabilité, c'est-à-dire que le système doit pouvoir être réutilisé dans des domaines d'application différents.

Dans ce contexte, certain développeurs proposent de partager des bases de données types que l'on pourra utiliser pour tester des différents projets liés à la réalisation de systèmes de fouille de données (accompagnés d'agents ou non). Ces sites sont souvent appelé « des fournisseurs de données synthetiques »

Fournisseurs de Bases de données Synthétiques :

Les fournisseurs de bases de données synthétiques sont comptés par centaines sur internet, chacun propose ses tables contenant des données ordonnées et typés (numériques, alphanumériques), respectant des structures diverses .

Notre préférence était de choisir l'un des fournisseurs les plus connus dans ce domaine d'application. Le site de « fimi » offre des structures de données synthétiques énormes et standardisées pour différents usages : http://fimi.ua.ac.be/data/

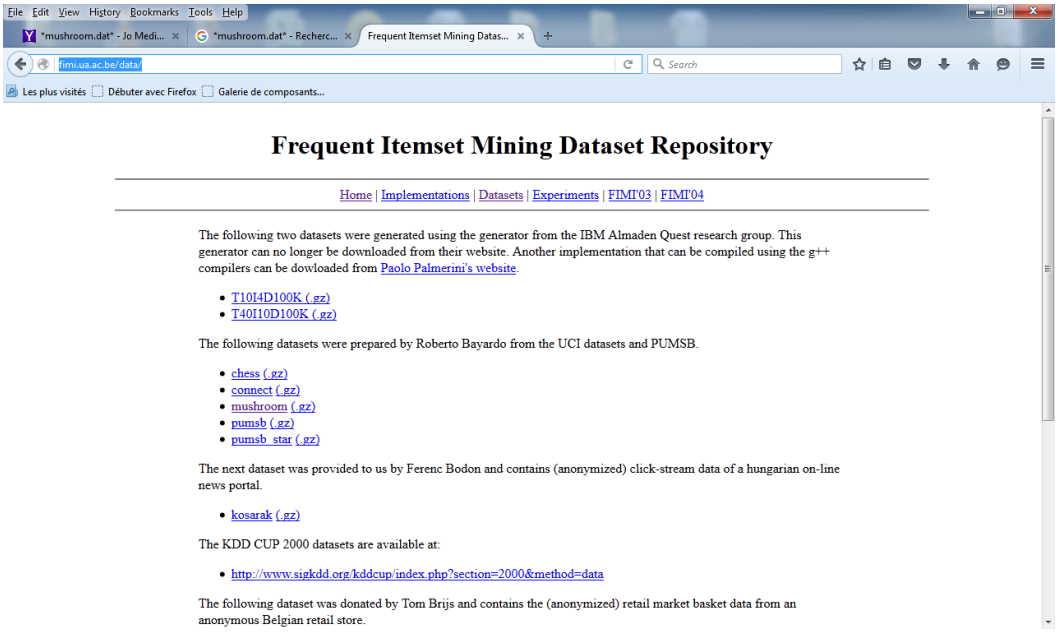


Figure 28 : Aperçu sur la page principale du site « fimi »

La structure de données la plus susceptible d'être sélectionnée pour notre suite de teste sur le système PADMA-RAD s'avère être le fichier « mushroom.dat », sa structure est bien ordonnée et son volume convient au type de traitement envisagé.

IV. Aperçu final sur la plateforme PADMA-RAD :

Figure 29 : Aperçu sur l'interface utilisateur du système PADMA-RAD

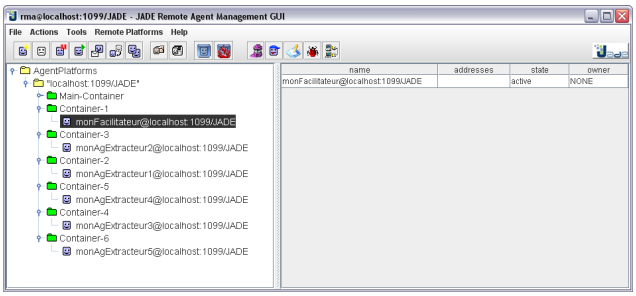


Figure 30 : L'IHM de la plateforme Jade accompagnant l'interface PADMA-RAD

Après la phase d'implémentation des différents éléments formant notre système, le lancement, on passe désormais au moment de la compilation et l'exécution de notre système bien achevé.

L'interface utilisateur de notre système s'exécute en mode fenêtré (voir figure 35), elle est constituée de trois boutons radio (JRadioButton) pour permettre à l'utilisateur de choisir le mode de fouille, on a un champ de texte(JTextField) pour y introduire un mot décrivant le sujet recherché et deux boutons (JButton) l'un deux permet de lancer la recherche et la fouille de données, l'autre bouton permet de quitter le programme (terminaison du système).

Lors du lancement de notre système, la fenêtre principale s'affiche (figure 35), si on choisi le mode de fouille distribué (choix2 ou choix3) on voit s'afficher la plateforme JADE en mode graphique (voir figure 36) et qui va accompagner le déroulement de notre processus d'exploration de données. On observe nos agents dans la plateforme chacun dans son conteneurs et on peut suivre leurs activités et les contrôler. pour plus de détail veuillez voir les annexes (Annexe02).

V. Test Opérationnel :

V.1-Introduction :

Cette section décrit la phase finale de notre travail, c'est là où on pourra évaluer la performance de notre système en comparant les différents modes d'exécutions de notre processus de fouille de donnée. C'est ici qu'on va qualifier la contribution des agents et des algorithmes distribués vis-à-vis des processus de fouille de données.

V.2-Lancement du test final du système :

V.2.1-Expérimentation ciblée (sur 05 tables pré-disposées):

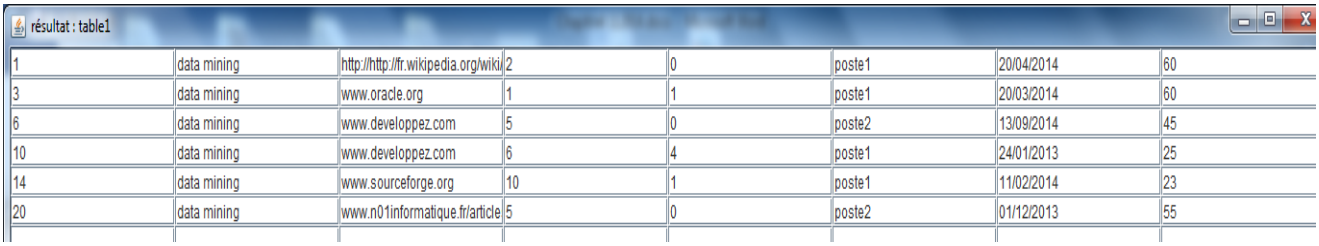
La phase du teste final consiste à introduire un mot (ex : data mining) dans le champ de texte et cliquer sur le bouton « valider », on assiste au lancement d'un processus de consultation de base de données suivi d'une extraction des motifs fréquents. Quand l'opération touche à sa fin on note la durée de l'exécution et le résultat obtenu après l'extraction.

On va utiliser le même mot clés pour les trois modes opératoires, on aura le même ensemble de résultats du point de vue de la recherche SQL, mais c'est la durée d'exécution et la précision du résultat de l'extraction des règles d'association qui nous intéresse.

V.3-Comparaison des résultats :

V.3.1-Expérimentation ciblée (à partir des tables pré-disposées) :

Après lancement de la recherche, la recherche SQL a retourné les résultats correspondant aux cinq tables modèles comme décrit dans la figure :


résultat : table1

1	data mining	http://http://fr.wikipedia.org/wiki/	2	0	poste1	20/04/2014	60
3	data mining	www.oracle.org	1	1	poste1	20/03/2014	60
6	data mining	www.developpez.com	5	0	poste2	13/09/2014	45
10	data mining	www.developpez.com	6	4	poste1	24/01/2013	25
14	data mining	www.sourceforge.org	10	1	poste1	11/02/2014	23
20	data mining	www.n01informatique.fr/article	5	0	poste2	01/12/2013	55

Figure 31 : Résultat SQL après consultation des tables sur PADMA-RAD

Voir plus de détails dans les annexes (Annexe 03)

a) Première Expérience : Mode sans technologie distribuée

Figure 32 : le côté apparent de l'IHM dans la première méthode de fouille

```
//************** mode sans agents
//**************
items Size 1 : poste1 - support :4
 items Size 1 : 60 - support :2
 items Size 1 : poste1 - support :2
 items Size 1 : www.developpez.com - support :2
 items Size 1 : poste2 - support :2
 items Size 2 : poste1 , www.developpez.com - support :1
 items Size 2 : 60 , poste1 - support :2
 items Size 2 : poste1 , www.developpez.com - support :1
 items Size 2 : www.developpez.com , poste2 - support :1
//**************
longueur liste condidats : 1
 items Size 1 : poste2 - support :2
//**************
longueur liste condidats : 1
 items Size 1 : poste1 - support :2
//**************
longueur liste condidats : 1
 items Size 1 : poste1 - support :2
//**************
longueur liste condidats : 1
 items Size 1 : poste5 - support :2
 //**************

 Total elapsed time in execution : 5183 millisec
```

b) Deuxième Expérience : Utilisation des agents (Algorithme Apriori classique)

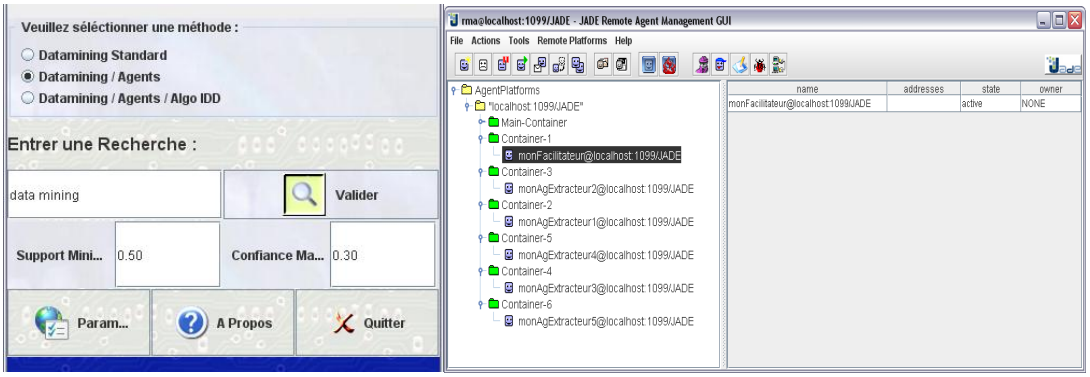


Figure 33 : le côté apparent de l'IHM dans la deuxième méthode de fouille

Agent1	Agent2	Agent3	Agent4	Agent5
items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : poste1 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : www.developpez.com , poste2 - support :1 //********** items Size 1 : poste2 - support :2 //********** items Size 1 : poste1 - support :2 //********** items Size 1 : poste1 - support :3 //********** items Size 1 : poste5 - support :2	items Size 1 : poste2 - support :2 //********** items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : poste1 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : www.developpez.com , poste2 - support :1 //********** items Size 1 : poste1 - support :2 //********** items Size 1 : poste1 - support :3 //********** items Size 1 : poste5 - support :2	items Size 1 : poste1 - support :2 //********** items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : poste1 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : www.developpez.com , poste2 - support :1 //********** items Size 1 : poste2 - support :2 //********** items Size 1 : poste1 - support :3 //********** items Size 1 : poste5 - support :2	items Size 1 : poste1 - support :3 //********** items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : poste1 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : www.developpez.com , poste2 - support :1 //********** items Size 1 : poste2 - support :2 //********** items Size 1 : poste1 - support :2 //********** items Size 1 : poste5 - support :2	items Size 1 : poste5 - support :2 //********** items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : poste1 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : www.developpez.com , poste2 - support :1 //********** items Size 1 : poste2 - support :2 //********** items Size 1 : poste1 - support :2 //********** items Size 1 : poste1 - support :3
5168 millisec	**5175 millisec**	**5171 millisec**	**5181 millisec**	**5173 millisec**

Tableau 7 : Résultat de la fouille en mode Multi agents (Méthode 02)

c) Troisième Expérience : l'utilisation des agents et de l'algorithme IDD

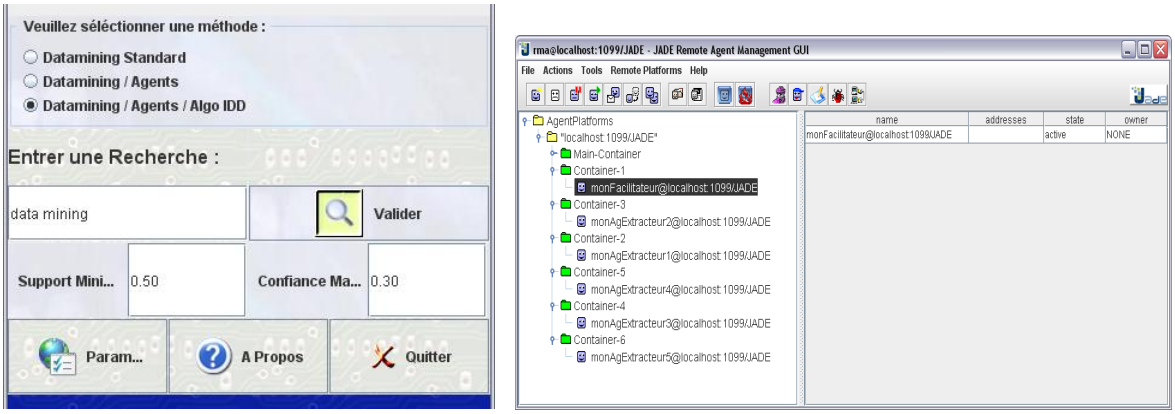


Figure 34 : le côté apparent de l'IHM dans la troisième méthode de fouille

Agent1	Agent2	Agent3	Agent4	Agent5
items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 1 : poste5 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : www.developpez.com , poste2 - support :1	items Size 1 : poste5 - support :2 items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : poste1 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : www.developpez.com , poste2 - support :1	items Size 1 : poste2 - support :2 items Size 1 : poste5 - support :2 items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : poste1 - support :2 items Size 1 : www.developpez.com - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : www.developpez.com , poste2 - support :1	items Size 1 : poste1 - support :4 items Size 1 : poste5 - support :2 items Size 1 : 60 - support :2 items Size 1 : poste1 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : www.developpez.com , poste2 - support :1	items Size 1 : poste5 - support :2 items Size 1 : poste1 - support :4 items Size 1 : 60 - support :2 items Size 1 : www.developpez.com - support :2 items Size 1 : poste2 - support :2 items Size 2 : poste1 , www.developpez.com - support :1 items Size 2 : 60 , poste1 - support :2 items Size 2 : www.developpez.com , poste2 - support :1
5099 millisec	**5102 millisec**	**5100 millisec**	**5101 millisec**	**5105 millisec**

Tableau 8 : Résultat de la fouille en mode Multi agents (Algorithme IDD)

V.3.2-Expérimentation non ciblée (Fichier : mushroom.dat):

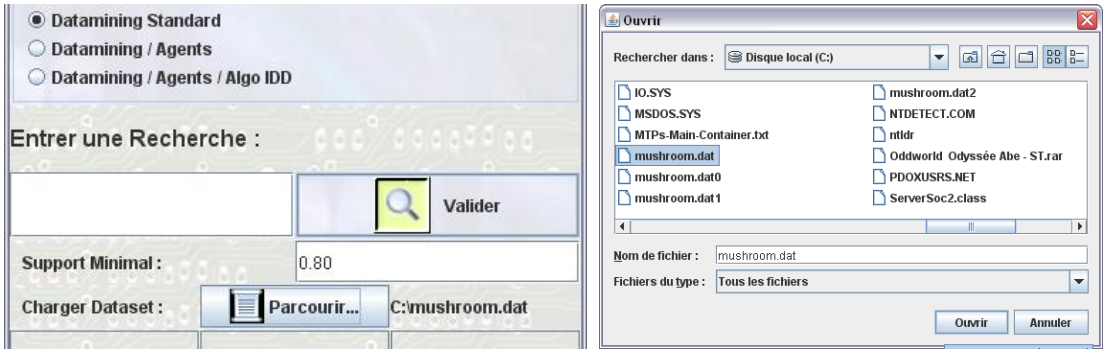


Figure 35 : Chargement du fichier mushroom.dat pour le test non ciblé

Après le chargement de la table « mushroom.dat », on clique sur le bouton « Valider » , cette méthode n'a pas besoin de saisir un mot pour la recherche car les données sont déjà classifiées et triées dans la table, on obtient comme dans les expériences précédentes une longue liste d'items et l'affichage d'une durée d'exécution selon la méthode choisie (sans Agents, avec Agents, avec algorithme IDD). (voir Annexe03)

V.4-Discussion des Résultats :

Pour la première expérience, le système procède sans avoir recours aux agents ni à l'algorithme distribué IDD, on a obtenu un ensemble de motifs fréquents en explorant les Data Sets un après l'autre de manière séquentielle, le lapse de temps nécessaire à cette opération est T=5183 millisec ayant en sortie un ensemble de 13 motifs fréquents pour un seul processus.

Cependant, la deuxième expérience, on a fait appel à la technologie des agents, chacun d'eux a exécuté sa propre recherche SQL et son propre algorithme d'extraction, ici l'algorithme d'extraction des données est « APRIORI », le résultat obtenu ne varie pas beaucoup d'un agent à un autre car la taille des Datasets de départ n'est pas volumineuse, la réponse est envoyée par les agents dans les délais respectifs : T1= 5168 millisec, T2= 5175 millisec, T3= 5171 millisec, T4= 5181 millisec, T5= 5173 millisec. Où chaque agent présente le même nombre d'items fréquents(13) mais dans un ordonnancement varié (voir Tableau 7). La légère baisse dans le temps d'exécution du processus de fouille revient au fait que l'algorithme APIORI est très adapté à ce genre d'architecture (PADMA-RA) et grâce à lui les agents s'échangent les données entre eux au lieu de parcourir tous les Dataset's en manière successive.

La troisième expérience, en combinant la technologie multi agents et l'algorithme IDD, on a obtenu un résultat remarquable, la réponse est envoyée par les agents dans les délais respectifs : T1= 5099 millisec, T2= 5102 millisec, T3= 5100 millisec, T4= 5101 millisec, T5= 5105 millisec. Ce qui signifie une une grande baisse dans le temps d'exécution des processus Data mining, cela est du à l'étape de partitionnement de l'algorithme IDD qui aide à réduire les transactions entre les agents, une fois le partitionnement est effectué dans notre cas, le temps d'échange entre les agents se trouve très réduit. De plus, la phase du partitionnement permet de reclasser les items et de supprimer les redondances, cela conduit également à réduire le temps de traitement et en plus donner un résultat plus distinct et plus précis.

Les données concernant les délais d'exécution minimaux des trois expériences ont été regroupées dans un même graphe pour les mettre en évaluation (voir figure 36 et 37) ,on a fait de même pour les itemsets obtenus dans les différents essais(voir figure 36 et 37).

Type de Processus	Nb items/proc	Temps maximum
Methode 1	13	5183
Methode 2	13	5175
Methode 3	8	5102

Tableau 9 : Les méthodes de fouille et leurs résultats respectifs (E. ciblée)

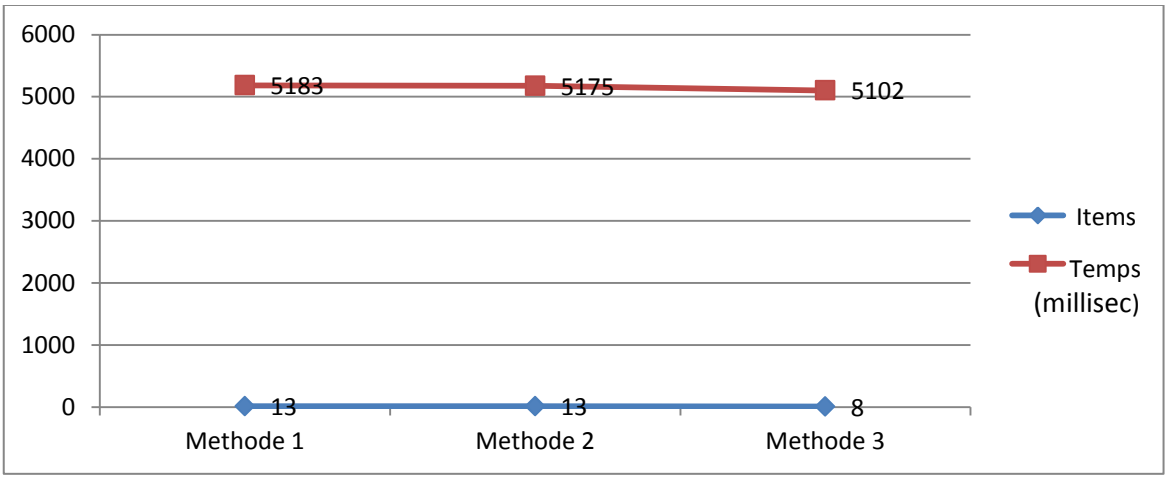


Figure 36 : le graphe d'évaluation des données de « Tableau9 »

Type de Processus	Nb items/proc	Temps maximum
Methode 1	476	56422
Methode 2	435	51109
Methode 3	347	50983

Tableau 10 : Les méthodes de fouille et leurs résultats respectifs (Expérimentation non ciblée)

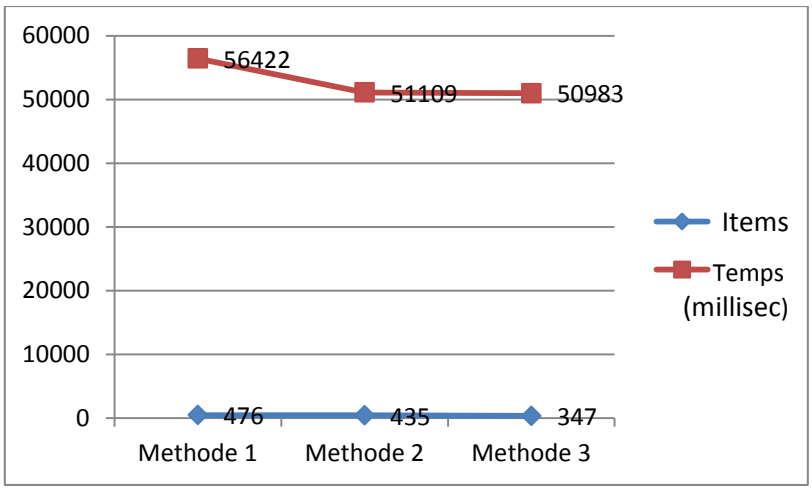


Figure 37 : le graphe d'évaluation des données de « Tableau10 »

Les graphes ci-dessus résument le résultat final, on note une diminution en temps d'exécution et une augmentation de précision exprimé ici par une diminution du nombre d'items fréquents retrouvés lorsqu'on procède à une fouille de données doté d'un système multi agents et d'un algorithme d'extraction des règles d'association distribué.

Tout ça, nous conduit à déduire que dans une architecture comme la nôtre (PADMA-RAD), la contribution des agents dans les processus de fouille de données s'est déclarée très positive ; les agents et la technologie distribuée ont joué un rôle majeur dans l'acquisition du système en vitesse et en performance.

Observations :

Les résultats présentés ici sont obtenus à partir de la plateforme PADMA-RAD, selon des conditions d'exécution optimales, c'est-à-dire, si ces conditions changent les résultats ne peuvent garder leur qualité, les troubles qui peuvent influencer l'efficacité de la plateforme sont :

- Les problèmes liés à la transmission réseau : si une ou plusieurs tables sont inaccessibles,
- Environnement d'exécution relativement lent (machine figée ou en panne) .
- Mauvaise configuration de la plateforme.

Si l'un des trois problèmes se présentent, les agents extracteurs vont être bloqués ou entrainés en boucle infinie, ce qui va empêcher le facilitateur de regrouper les données pour représenter le résultat final dans un bon délai, et le temps d'exécution va augmenter, donc il y aura perte de temps.

Conclusion :

Dans ce chapitre, nous avons décrit le processus d'implémentation, le teste et l'évaluation des performances de notre système qu'on a baptisé PADMA-RAD, nous avons aussi expliqué qu'on devait le diviser en 3 niveaux pour faciliter l'étape du teste fonctionnel et de l'évaluation des performances. Pendant la dernière étape, les résultats ont montré que grâce à la technologie des agents et des algorithmes distribués, le système a gagné en rapidité et en précision, malgré la taille restreinte des données prises en échantillon et le nombre réduits d'agents, ces deux caractéristiques (rapidité et précision) ont démontré une grande performance du système, qu'on pourra présenter comme solution lorsqu'on fera face à un processus de fouille de données en mode distribué.

Conclusion Générale

-Conclusion

-Résumé

Conclusion

Cette thèse regroupe des analyses dans deux domaines algorithmiques bien distincts, on a d'un côté les techniques utilisées dans la fouille de données (Data Mining) dont l'algorithme « APRIORI » fait partie, on a d'un autre côté la technologie des systèmes multi agents et les solutions algorithmiques distribuées. Nous avons déjà expliqué chacune de ces disciplines du point de vue méthodologique dans le contexte du travail demandé.

Bien que cette thèse soit partagée en plusieurs parties, mais chacune d'elles définit une démarche pour aboutir au travail final, ce travail qui consiste à créer un système combinant une architecture multi agent (PADMA) et un système de fouille de données basé sur l'extraction des règles d'association et ayant pour modèle l'algorithme « APRIORI » pour donner naissance à une autre discipline (la fouille distribuée).

Notre travail repose sur l'étude et l'évaluation du système obtenu, c'est à dire étudier la contribution des systèmes multi agents dans une telle combinaison de technologies et vérifier les avantages qui peuvent être apportés dans un système comme « PADMA-RAD » pour savoir si on peut le définir en tant que solution méthodologique aux problèmes de fusion entre SMA et Data Mining.

Cette thèse ne propose qu'une étude modeste parmi d'autres, c'est un argument de plus pour les futurs chercheurs qui auront l'occasion d'étudier des sujets dans le même contexte.

Résumé

L'objectif de ce travail est d'apporter une réponse concernant la contribution des systèmes multi-agents dans les processus de fouille de données distribuées. Dans une première partie nous étudiant les projets les plus connus, qui sont les fruits de la fusion des technologies des SMA et le Data Mining de type DMBA (Data Mining Basé sur les Agents). En second lieu nous avons étudié les architectures des systèmes DMBA afin de choisir un modèle sur lequel on battit notre propre système (PADMA), dans la troisième étape nous avons visité les algorithmes distribués les plus utilisés dans des contextes similaires au notre, dont l'algorithme IDD fait partie. Enfin, la dernière partie est dédiée à la mise en œuvre et à l'évaluation de notre système achevée (PADMA-RAD) afin de pouvoir répondre à la problématique de la thèse.

Mots clés : Algorithmique ; structures de données ; Bases de données; Fouille de données ; Règle d'association ;Data Mining ; Architecture Parallèle ; Agent .

Abstract

The objective of this work is to provide an answer concerning the contribution of multi-agent systems in distributed data mining process. In the first part we studying the best-known projects, which are the fruits of the merger of SMA technology and type of DMBA Data Mining (Data Mining Based on Agents). Secondly we studied the architecture of DMBA systems to choose a model on which our system is defeated (PADMA), in the third step we visited the distributed algorithms commonly used in similar contexts to our own, whose IDD algorithm belongs. The last part is dedicated to the implementation and evaluation of our complete system (PADMA-RAD) in order to address the problem of the thesis.

Keywords: Algorithms, data structures; Databases; Data mining; Association rule; Data Mining; Parallel architecture; Agent

Références

-Réferences

-Glossaire

-Annexes

References:

Bibliographies :

[01] "**Evolution of Database Technology**": **Databases and *Data Mining***
Projects at LIACS. By EM Bakker, Introduction and Database Technology,
April 2005, http://www.liacs.nl/~erwin/.../02_dbdm2012_databases.pdf

[02] "**« DATA MINING » ou FOUILLE DE DONNÉES**": **Databases and *Data Mining***
By Gilbert Saporta, Décembre 2004,
https://cedric.cnam.fr/fichiers/RC1034.pdf

[03] "**Scalable, Distributed Data Mining – An Agent Architecture**": **Computational Science Methods**
Group, By Hillol Kargupta and Ilker Hamzaoglu and Brian, Décembre 1997,
https:// www.csee.umbc.edu/~hillol/PUBS/padmaKDD.pdf

[04] Support de Cours « Data Mining , fouille de données: Concepts et techniques »,
Faculté de Médecine de Marseille. By Marius Fieschi, Février 2006,
ttp://cybertim.timone.univ-mrs.fr/enseignement/doc-enseignement/informatique/introdatawarehouse/docpeda_fichier

[05] "**Apprentissage supervisé**", TELECOM ParisTech**.** By Fabrice Rossi, Mai/Juin 2009,
http://*apiacoa.org/publications/teaching/data-mining/supervised.pdf*

[06] "Apprentissage Non supervisé", Y. Bengio, J-F. Paiement, and P. Vincent. Out-of-Sample
Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. Technical Report
1238, Département d'informatique et recherche opérationnelle, Université de Montréal,
2003. http ://*www.iro.umontreal.ca/~pift6266/A06/cours/unsupervised.pdf*

[07] "Data Mining Distribué", DISDAMIN: Algorithmes de Data Mining Distribués
Bernard TOURSEL, Valerie FIOLET, Equipe PALOMA -LIFL -USTL -LILLE (FRANCE),
Clermont Ferrand -Janvier 2003
http://www-sop.inria.fr/axis/fdc-egc04/Slides/Fiolet-Tourcel.pdf

[08] "Support de cours : Algorithmes d'Extraction de Régles d'Association", Thierry Lecroq
Université de Rouen FRANCE, 2010.
http ://www-igm.univ-mlv.fr/~lecroq/cours/regles.pdf

[09] "Support de cours : d'Extraction de Régles d'Association- Chapitre1", Maria Malek
Institut EISTI FRANCE, 2009.
http:// www.eisti.fr/~mma/HTML-IA/Cours/Cours4/ia03.pdf

[10] "Systèmes Multi Agents : Introduction Aux Systèmes multi-agents**",** Julien Saunier
Institut IFSTTAR FRANCE, Septembre 2009.
http:// www.lamsade.dauphine.fr/~saunier/sma/SMA-2013-intro.pdf

[11] " Systèmes Multi Agents ", vers une intelligence collective
Jacques Ferber, Edition IIA, INTEREDITION 1995
http://www.lirmm.fr/~ferber/publications/LesSMA_Ferber.pdf

[12] " Data Mining and Multi-agent Integration ", Faculty of Engeneering and Information
By Longbing Lao, Sydney university, Springer 2007
http://www-staff.it.uts.edu.au/~lbcao/

[13] "MULTI AGENT-BASED DISTRIBUTED DATA MINING: AN OVER VIEW ", Research Scholar,CSIT
Department,JNT University , Hyderabad. Andhra Pradesh, INDIA , 2010
http://*www.ijric.org/volumes/Vol3/11Vol3.**pdf***

[14] "Les systèmes de détection d'intrusion basés sur du machine learning", By Liran LERMAN,
UNIVERSITÉ LIBRE DE BRUXELLES, Faculté des Sciences, Département d'Informatique,
Bruxelles 2009.
http://student.ulb.ac.be/~llerman/detectionDIntru/etatDeLartV4.pdf

[15] "Java Agent Developpement Environnement : JADE", Developing Multi-Agent Systems with JADE
Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood
EDITION WILEY Mars 2007,
homepages.abdn.ac.uk/w.w.vasconcelos/teaching/.../jade_book.pdf

[16] "Fast Algorithms for Mining Association rules in large Databases",R. Agrawal, R.Srikant
Proc. VLBD conf. September 1994

[18] "ADVANCES IN KNOWLEDGE DISCOVERY AND DATA MINING",
U.M.Fayyad, G. Piatetsky, P. Smith and R. Uthurusamy
KLUWER ACADEMIC PUBLISHER, AAAI Press 1996,

Webographie :

[01W] Data mining : http://www.rithme.eu/?m=resources&p=dmdomains&lang=fr
[02W] Apprentissage Automatisé : *http://www.univ-tlemcen.dz/~benmammar/IA2.pdf*
[03W] Data Mining Distribué : http://ori.univ-lille1.fr/notice/view/univ-lille1-ori-2364
[04W] Algorithmes d'Extraction de Données distribués :
http://www.academia.edu/1621200/Application_de_K-means_%C3%A0_la_d%C3%A9finition_du_nombre_de_VM_optimal_dans_un_cloud
[05W] Règles d'Association : http :// *www.csis.pace.edu/~scharff/DMIFI/associations5.ppt*
[06W] Algorithm APRIORI : http://blog.khaledtannir.net/2011/05/apriori/#.VIg3kWdrZ6g
[07W] PAPYRUS1 : http:// *www.csee.umbc.edu/~hillol/PUBS/review.pdf*
[08W] PAPYRUS2 : http:// *papers.rgrossman.com/proc-053.pdf*
[09W] JAM1 : http:// *www.csee.umbc.edu/~hillol/PUBS/review.pdf*
[10W] JAM2 : http:// *www.cs.columbia.edu/techreports/cucs-007-01.ps*
[11W] PADMA1 : http:// *www.csee.umbc.edu/~hillol/PUBS/review.pdf*
[12W] PADMA2 : *eric.univ-lyon2.fr/~pkdd2000/Download/T1.pdf*
[13W] Algorithmes Distribués : https://globaljournals.org/GJCST_Volume13/5-Efficient-Distributed-Algorithm.pdf
[14W] JADE Tutorial1 : http://jade.tilab.com/doc/tutorials/JADEAdmin/startJade.html
[15W] JADE Tutorial2 : http://djug.developpez.com/java/jade/creation-agent/
[16W] JAVA APRIORI EXEMPLE : https://gist.github.com/monperrus/7157717
[17W] Divers E-Book : https://www.archive.org/
[18W] Tutoriels en Vidéo : https://www.Youtube.com

Mémoires et Thèses :

[01M] **Mémoire de thèse de Doctorat "** Quelques Problèmes d'Apprentissage Supervisé et Non Supervisé » , By Thomas Laloé, Spécialité Mathématiques appliquées, L'UNIVERSITE MONTPELLIER II, Novembre 2009,
*https://hal.archives-ouvertes.fr/tel-00455528/**PDF**/thesepois.**pdf***
[02M] **Mémoire de thèse de Master «** les Règles d'Associations distribuées : équilibrage des charges et Migration des items» , par Melle Kebbati Nadjia, informatique, USTO, Année 2009/2010,
[03M] **Mémoire de thèse de Doctorat «** l'Algorithme PGCD et la Fouille de données : Analyse dynamique» , par Loïck Loth, informatique, L'UNIVERSITE de Caen (Normandie) , Année 2009/2010,
https://lhote.users.greyc.fr/articles/these-loick.ps

GLOSSAIRE:

Tableau des Abréviations :

Abréviation	***Sighification***
CCPD	Common Condidate Partitionned Database
CD	*Count Distribution*
DD	*Data Distribution*
DM	*Data Mining*
DDM , DMD	*Distributed Data Mining (Data Mining Distribué)*
DMBA	*Data Mining Basé sur les Agents*
ECD	*Extraction de Connaissance à partir des Données*
GUI	*Graphic User Interface*
HD	Hybride distribution
HPA	Hash Partioned Apriori
I/O	*Input and Output (Entrée et Sortie)*
IDD	Intelligent Data Distribution
IDE	Interfaced Developpement Environnement
JADE	Java Agents Developpement
JAM	*Java Agents for Metalearning*
JEE	*Java Entreprise Edition*
KDD	*Knowledge Discovery from Data*
MAD-IDS	*Multi Agent Datamining – Intrusion Detection System*
SMA	*Systèmes Multi Agents*
SBuff	*Send Buffer*
PAPYRUS	*a system for data mining over local and wide area clusters and super-clusters*
PADMA	*PArallel Data Mining Agents*
RA	*Règles d'Association*
RAD	*Règles d'Association Distribuées*
RBuff	*Read Buffer*

ANNEXES:

Annexe01 : *Extraits du Code Source du Projet PADMA-RAD*

A1- Voici les étapes à suivre pour installer JADE : [15,14W,18W]

2. téléchargez le fichier JADE-all-3.6.zip de l'adresse suivante : http://jade.tilab.com/download.php
3. décompressez le fichier (on va supposer tout au long de ce tutorial que le chemin du répertoire JADE-all-3.6 est le c:\ JADE-all-3.6). Apres avoir décompressé le fichier vous retrouvez quatre autres fichiers ZIP (JADE-bin-3.6.zip , JADE-doc-3.6.zip, JADE-examples-3.6.zip, JADE-src-3.6.zip). Décompressez ces 4 fichiers
4. on doit maintenant mettre à jour la variable **classpath** (si elle n'existe pas encore il faut la créer) En faisant comme suit :

   a. par Clic droit sur le poste de travail, choisissez propriétés. La fenêtre propriétés système apparaît , choisissez l'onglet Avancé Puis cliquez sur variables d'environnement
   b. créer une variable d'environnement intitulée CLASSPATH indiquent l'emplacement des fichiers propres à la plateforme JADE tel que décrit dans la figure suivante.

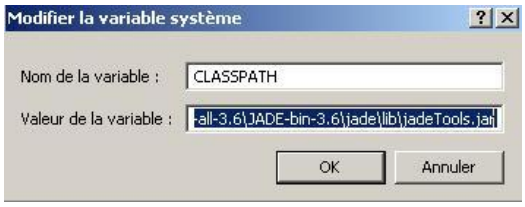


Figure : Aperçu d'écran illustrant la modification de variable d'environnement

A2- Voici les étapes à suivre pour configurer JADE : [15,14W,18W]

Sélectionnez

```
Java jade.Boot -gui
```

Une fenêtre dos s'ouvre qui lance la plateforme jade. La fenêtre suivante apparaît :

Figure : Aperçu d'écran illustrant la plateforme JADE en mode graphique

A3- Voici les étapes à suivre pour créer son premier agent avec JADE :

Nous, allons créer notre premier agent. En utilisant un, modèle fourni avec la plateforme situé dans : C:\JADE-all-3.6\JADE-examples-3.6\jade\src\examples\hello. Ouvrez eclipse et créez un nouveau projet (MyFirstAgent par exemple), ajoutez un package (firstAgent) puis créez une nouvelle classe appelée HelloWorldAgent. écrire le code suivant dans la classe. [15,14W,18W]

```
package pkg;
import jade.core.Agent;

public class myAgent extends Agent {

  protected void setup() {
        System.out.println("Hello World! My name is "+getLocalName());

  }
}
```

Vous remarquez l'existence de plusieurs erreurs dans ce petit code. Pour résoudre ce petit problème,effectuez un clic droit sur le nom du projet Puis choisissez propriétés. Cliquez sur java build path >> Libraries>> add external JARs (voir la Figure ci dessous)

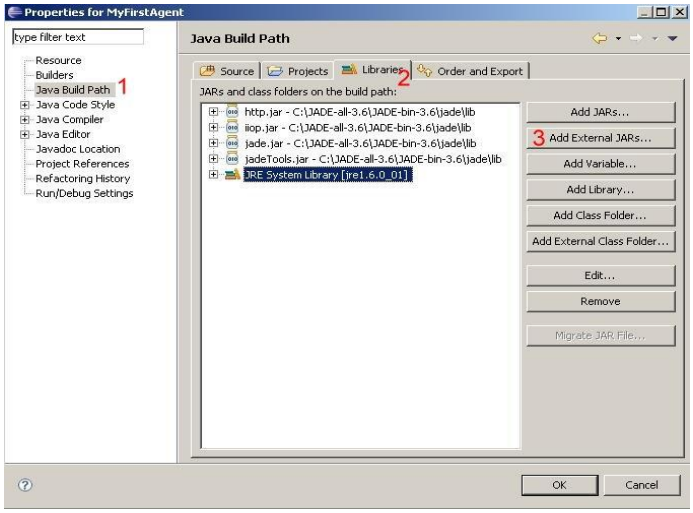


Figure : Aperçu d'écran illustrant la configuration du java build path

Ajoutez les fichiers .jar situés dans C:\JADE-all-3.6\JADE-bin-3.6\jade\lib puis cliquez sur Ok. Il reste à compiler et lancer l'agent pour cela :
-Allez dans run>>Run configuration
-double-cliquez sur java application Dans l'onglet " main "
-dans la zone de saisie Main class, tapez le code suivant : jade.Boot
-cochez la case : " Include librairies when searching for a main class "

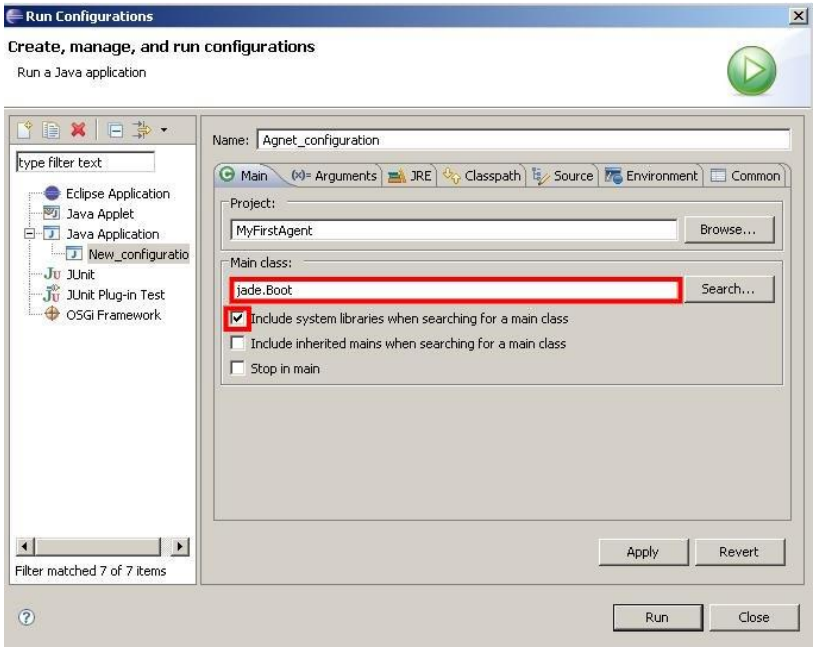


Figure : Aperçu d'écran illustrant la configuration de la classe de l'agent JADE

Et dans l'onglet arguments, tapez le code suivant :
-gui jade.Boot NomDuL'agent:LeNomDuPackage.LeNomDeLaClasse
Dans notre exemple on tape le code suivant :
-gui jade.boot smith:pkg.myAgent
puis cliquez sur « apply » pour ne pas refaire cette configuration plusieurs fois dans le même projet

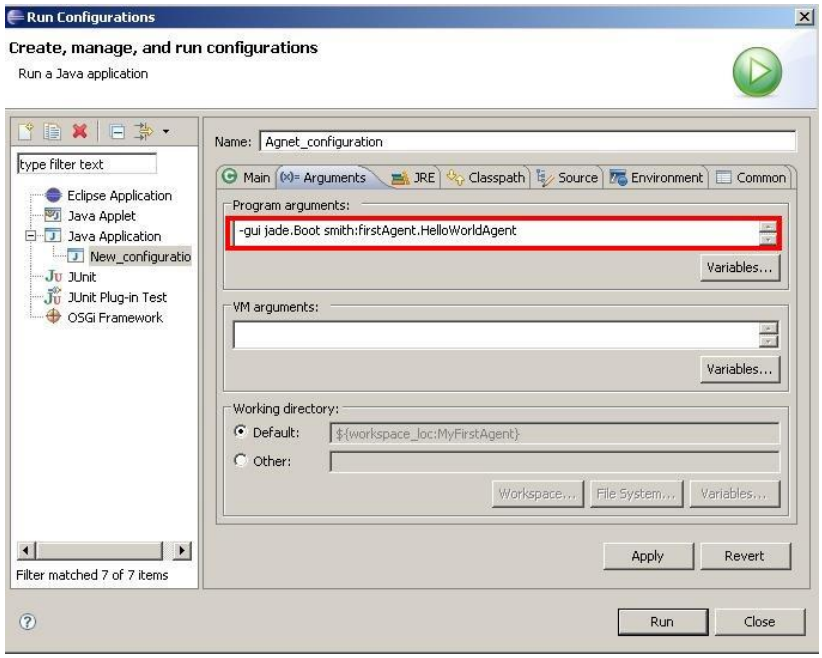


Figure : Aperçu d'écran illustrant la commande de démarrage de l'agent JADE

Cliquez sur « run » pour voir le résultat (affichage de Hello World! My name is smith Dans la console et ouverture du la plateforme jade) .

A4- Notions sur les Conteneurs de la plateforme JADE : [15,14W,18W]

Lorsqu'on lance la plateforme jade soit en mode commande ou en mode graphique, on s'aperçoit qu'elle crée des espaces logiques appelés conteneurs (containers), chaque conteneurs peut représenter un espace mémoire, un site local ou distant. Tous les conteneurs sont contrôlés par un conteneur principal.

Le conteneur principal à l'origine contient trois agents mis en place par la plateforme, ils sont indispensables à la gestion des autres conteneurs qui contiendront à leur tour des agents.

Dans notre cas, on a six conteneurs à mettre en place, chacun d'eux doit représenter un site, le premier va représenter le site du gestionnaire du réseau où on va créer notre interface utilisateur et notre agent facilitateur (en mode distribué), les cinq autres conteneurs joueront le rôle des dépôts de données où on va créer les agents de Fouille (les agents extracteurs).

A5- Implémentation des Conteneurs de la plateforme JADE :

a-Création d'un Conteneur principal :

```
import jade.core.ProfileImpl;
import jade.core.Runtime;
import jade.util.ExtendedProperties;
import jade.util.leap.Properties;
import jade.wrapper.AgentContainer;
import jade.wrapper.ControllerException;

. . .

        try {
                Runtime rt=Runtime.instance();
                Properties p=new ExtendedProperties();
                p.setProperty("gui", "true");
                ProfileImpl pc=new ProfileImpl(p);
                AgentContainer ac=rt.createMainContainer(pc);
        } catch (Exception e) {
                e.printStackTrace();
        }

. . .
```

Explication :

Ce module permet de faire appel au Conteneur principal de la plateforme JADE, il est nécessaire si on décide de lancer le mode multi-agents de notre système. C'est grâce à ce conteneur principal de la plateforme que seront créés les autres conteneurs qui joueront le rôle de sites pour héberger nos futurs agents.

Le conteneur principal de la plateforme Jade contient à son tour des agents prédisposés pour remplir des tâches nécessaires au fonctionnement de la plateforme elle-même et aussi aux système conçu sur sa base.

b-Création d'un Conteneur non principal :

```
import jade.core.ProfileImpl;
import jade.core.Runtime;
import jade.util.ExtendedProperties;
import jade.util.leap.Properties;
import jade.wrapper.AgentContainer;
import jade.wrapper.AgentController;
import jade.wrapper.ControllerException;

. . .

        try {
                        Runtime rt=Runtime.instance();
                        ProfileImpl pc=new ProfileImpl(false);
                        pc.setParameter(ProfileImpl.MAIN_HOST, "localhost");
                        AgentContainer ac=rt.createAgentContainer(pc);
                        AgentController agc=ac.createNewAgent("monFacilitateur",
            "pkg.Facilitator", new Object[]{}); // conteneur conçu pour hébérger l'agent Facilitateur
                        agc.start();
                } catch (Exception e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

. . .
```

Explication :

Ce module permet de créer un Conteneur non principal dans la plateforme JADE, il est nécessaire pour abriter et servir de milieu d'exécution pour les agents de notre système. Un conteneur peut héberger un ou plusieurs agents à la fois, il peut représenter un site local (mode localhost) de la même façon qu'il peut représenter un site distant (en donne l'adresse IP), il est aussi nécessaire si on choisit le mode multi-agents de notre système.

A6- Les Comportements des Agents « Behaviours » :

Définition :

Un comportement ou Behaviour est une tâche que l'agent doit effectuer , on peut la lui attribuer grace à l'instruction addBehaviour(Behaviour b) ;
Il est composé de 4 méthodes :

- onStart() ;
- Action() ; / elle désigne l'opération effectuée
- Done() ;// elle retourne true si action () est terminée
- onEnd() ;

Les types de Behaviour :

One-shot Behaviour :
Classe jade.core.Behaviours.OneShotBehaviour
Il s'exécute une seule fois puis se termine, elle implémente la méthode done() qui va toujours retourner true

Cyclic Behaviour :
Classe jade.core.Behaviours.CyclicBehaviour
Il s'exécute de manière répétitive, elle implémente la méthode done() qui va toujours retourner false

Generic Behaviour :
Classe jade.core.Behaviours. Behaviour
Il vient entre les deux premiers, il accepte la personnalisation, il laisse le choix de terminaison au programmeur.

Waker Behaviour :
Classe jade.core.Behaviours.WakerBehaviour
Il s'exécute grâce à la méthode onWake() dans un délais ou une date en argument ;(comme un réveil)

Ticker Behaviour :
Classe jade.core.Behaviours.TickerBehaviour
Il s'exécute grace à la méthode onTick() dans un lapse de temps répétitif ;(comme un timer)

La création d'un Behaviour (Exemple):

Reprendre le code de l'agent Facilitateur:

```
Private String items;
Protected void setup(){   // creation
Object[] args=get argument();
If (args.length==1){
items=(String)args(0);
System.out.println("Agent Facilitateur : "+this.getAID()+"je veux récupérer un items :"+items);

addBehaviour(new TickerBihaviour(this,1000){ // this est lagent Facilitator , 1000 est 1000 milisec
private int compteur=0 ;
protected void onTick(){
        compteur++ ;
System.out.println("tentative d'envoi  N° "+ compteur) ;
}

}) ;

}else {
System.out.println("veuillez définir un item") ;
doDelete() ;
}
}
```

A7- La communication entre les Agents « ACL Message » :

Messages ACL :

ACL est une norme de transmission adaptée pour les SMA, mise au point par la FIPA-ACL (une Organisation pour la recherche des SMA). Package jade.lang.acl.

Envoyer un message

```
. . .
ACLMessage message1=new  ACLMessage(ACLMessage.INFORM) ;
message1.addReceiver(new AID(« vendeur1 »,AID.ISLOCALNAME)) ;
message1.setContent(« livre XML ») ;
// message1.setObject(monObjet) ;  pour envoyer des objets
send(message1) ;
. . .
```

recevoir et lire un message : créer un behaviour pour lire les messages

```
ACLMessage message2=receive() ;
If (message2 !=null){
    System.out.println("message reçu :"+message2.getContent) ;
} else {
Block(); // arrêter le behaviour de lecture
}
```

A8- La création d'une classe Thread et son Lancement :

```
//**************   Création de la classe Thread intitulé  « Agentx »
. . .
Public classe Agentx extends Thread{
…
Public Agentx(paramétres){
// instruction du constructeur
…
}
Public void run(){
//*****  ce que doit faire le thread
}
…
}
. . .
//**************  Lancement de   la classe Thread intitulé  « Agentx »
Agentx ag1=new  Agentx() ;
Ag1.start() ;
```

A9- Le Module de Préparation des données pour l'extraction :

```
//************** preparation des données avant processus de fouille
                        // res1 est du type ResultSet de la librairie java.sql.*
                boolean LaSuite1=res1.next();
                                System.out.println("debut rendu : table1");
                                while(LaSuite1){
                                        System.out.println("\n ligne "+(id1)+" : ");
                                        for (int i=1;i<nbCols1+1;i++){
                                                System.out.print("Col"+i+":
                    "+res1.getString(i)+" - ");
                                                matr1[i-1][id1]=res1.getString(i);
                                        }
                                        LaSuite1=res1.next();
                                        id1++;
                                }
                                res1.close();
                                System.out.println("fin du rendu : table1");
                        for (int x=0;x<30;x++){
                          c=0;
                          if (matr[0][x]!=null){
                                rcdcount++;// récupérer les données non érronées
                          }
                          for (int y=0;y<8;y++){
                                if ((y==0)||(y==2)||(y==5)||(y==7)){
                                        tableD[c][x]=matr[y][x]; // conversion des données
                                        c++;
                                }
                          }
                 }
. . .
```

Explication :

Ce bloc est un code répétitif permettant de traduire les données résultantes de la recherche SQL (ensemble de type ResultSet) en un ensemble de données homogènes prêt à subir un procédé d'extraction des règles d'association (ensemble de patterns ou items).

Dans notre cas l'ensemble de patterns obtenu est un tableau de String (TableD[][]) dont le nombre de ligne dépend du nombre d'enregistrements obtenus lors de la recherche SQL, le nombre de colonnes est de longueur k=4.

A10- Le Module de connectivité mySQL :

```
. . .
import java.sql.*;
. . .
String DBLoc="jdbc:mysql://localhost:3306/mag";
        String Error="";
        String rech="";
        String req1="";
        String title1[]=new String[8];
        String matr1[][]=new String[8][100];
. . .

try{
                        Class.forName("com.mysql.jdbc.Driver");
                        Connection c1=DriverManager.getConnection(DBLoc,"root","");
                        ResultSet res1=null;
                        //***************  table1
                        try{
                                req1="SELECT * FROM table1 WHERE Sujet ='"+rech+"'";
                                Statement st1=c1.createStatement();
                                res1=st1.executeQuery(req1);
                                ResultSetMetaData resm1=res1.getMetaData();
                                int nbCols1=resm1.getColumnCount();
                                for (int x=1;x<nbCols1+1;x++){
                                        title1[x-1]=resm1.getColumnName(x);
                                }
                                System.out.println("nombre de champs :"+nbCols1);
                                int id1=0;
                                boolean LaSuite1=res1.next();
                                while(LaSuite1){
                                        System.out.println("\n ligne "+(id1)+" : ");
                                        for (int i=1;i<nbCols1+1;i++){
                                                System.out.print("Col"+i+":
  "+res1.getString(i)+" - ");
                                                matr1[i-1][id1]=res1.getString(i);
                                        }
                                        LaSuite1=res1.next();
                                        id1++;
                                }
                                res1.close();
                                System.out.println("fin du rendu : table1");
. . .
```

Explication :

C'est l'un des bloc les plus utilisé dans notre système PADMA-RAD, il a pour mission de récupérer le mot recherché par l'utilisateur, se connecter à la base de données et lancer une requête SQL pour retourner un objet de type ResultSet qui va contenir les résultats de la recherche préliminaire et servir de base pour le

processus d'extraction de données. Ce bloc est utilisé dans tous les niveaux de notre système, il est également incorporé aux agents mineurs.

A11- Le Module d'extraction des règles d'association (Extrait) :

```
double suppmin=0.30; // support minimal définit par l'utilisateur

. . .

//**** etape  : recherche des condidats
                for (int a=0;a<rcdcount;a++){ // scan de l'ensemble des données

                        for (int b=1;b<4;b++){
                    item1=tableD[b][a];
                    supp=1;
                //*******************
                    for (int i=a+1;i<rcdcount;i++){
                        for (int ii=1;ii<4;ii++){
                     if (item1.equals(tableD[ii][i])){
                        supp++; // support de l'item courant
                        if ((supp*100/rcdcount)>=suppmin){
                                itemset[0][pos1]=item1; // ajout de l'item si son support >= suppmin
                                itemset[3][pos1]=Integer.toString(supp);// marquer le support courant
                                    pos1++;
                        }    }    }
                        }
                    //*********************
                        }
                }
                System.out.print("\n longueur liste condidats : "+pos1);
                //************  liste des candidats C1

. . .
```

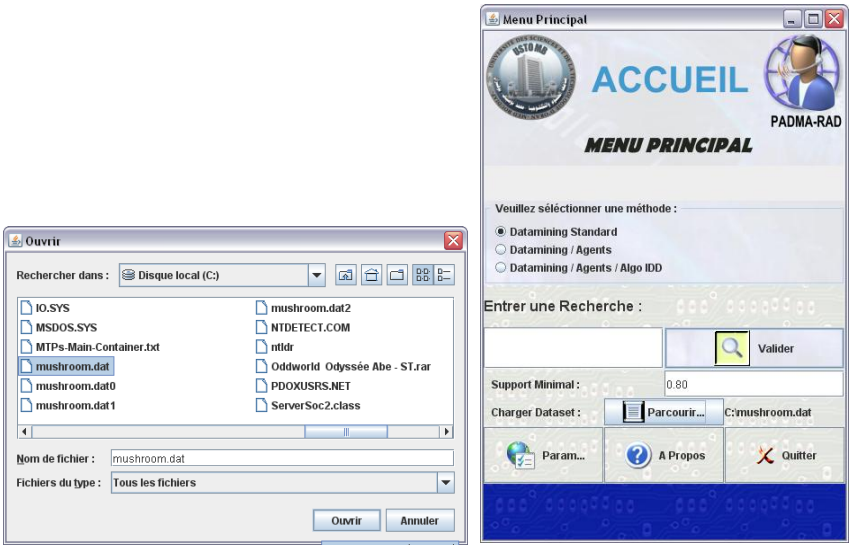
Explication :

Ce bloc est un code répétitif très complexe, on ne va présenter ici qu'un extrait. La séquence décrite dans cette rubrique fait partie de l'implémentation de l'algorithme APRIORI servant pour l'extraction des règles d'association en mode distribué, l'extrait désigne le bloc de recherches des items fréquents (les candidats locaux) de longueur C1=1 incluant le teste du support minimal.

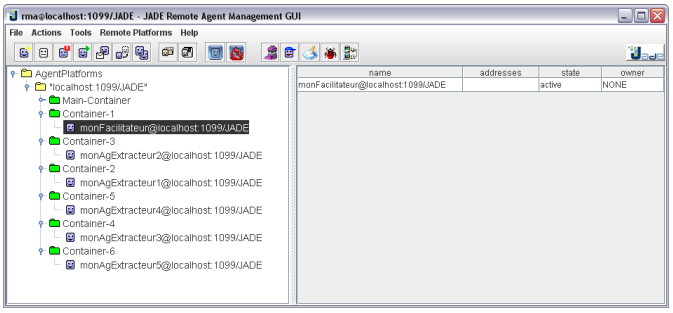
Annexe02 : *Aperçus sur l'interface graphique du projet PADMA-RAD*

Aperçus d'écran sur la fiche principal de l'IHM du système PADMA-RAD

Aperçus d'écran sur la fiche de paramétrage du système PADMA-RAD



Aperçus d'écran sur la fiche de paramétrage du système PADMA-RAD



Aperçus d'écran sur la fiche de paramétrage du système PADMA-RAD

Annexe03 : Aperçus sur les Résultats de consultation des Bases de Données

résultat : table5

1	data mining	www.wikipedia.org	5	2	poste1	11/06/2014	80
11	data mining	www.code-source.com	8	3	poste5	31/03/2014	95
15	data mining	www.developpez.com	12	5	poste5	10/10/2014	35

Aperçus sur les Résultats de consultation de la table Table5

résultat : table4

3	data mining	www.commentcamarche.org	6	1	poste1	11/10/2013	80
7	data mining	www.sourceforge.org	24	5	poste2	30/04/2014	66
10	data mining	www.oracle.org	6	6	poste1	20/08/2014	88

Aperçus sur les Résultats de consultation de la table Table4

résultat : table3

2	data mining	www.wikipedia.com	5	0	poste2	12/12/2013	74
5	data mining	www.commentcamarche.com	5	3	poste1	12/03/2014	55
8	data mining	www.developpez.fr	4	3	poste5	04/01/2014	76
13	data mining	www.stargamer.com	2	0	poste1	12/05/2014	30

Aperçus sur les Résultats de consultation de la table Table3

résultat : table2

3	data mining	www.oracle.org	5	1	poste2	30/01/2014	65
7	data mining	www.developpez;com	8	3	poste1	19/09/2013	50
11	data mining	www.commentcamarche.com	12	2	poste2	30/08/2014	78
16	data mining	www.code-source.com	3	2	poste5	10/10/2013	90

Aperçus sur les Résultats de consultation de la table Table2

résultat : table1

1	data mining	www.wikipedia.org	2	0	poste1	12/03/2014	50
3	data mining	www.oracle.org	1	1	poste1	20/03/2014	60
6	data mining	www.developpez.com	5	0	poste2	13/09/2014	45
10	data mining	www.developpez.com	6	4	poste1	24/01/2013	25
14	data mining	www.sourceforge.org	10	1	poste1	11/02/2014	23
20	data mining	www.n01informatique.fr/article	5	0	poste2	01/12/2013	55

Aperçus sur les Résultats de consultation de la table Table1

Aperçus sur les Résultats de l'extraction des règles d'association