

Université Saâd DAHLAB de Blida**Faculté des sciences****Département d'Informatique**

Mémoire présenté par :

M^{lle} HEBIB WARDA et M^r GHEBRIOUT YUCEF

Pour l'obtention du diplôme de Master

Domaine : Mathématique et Informatique**Filière : Informatique****Spécialité : Ingénierie des logiciels**

Sujet :

**Suivi des soldes dans le système ATCI à travers les
échanges interbancaires :**

Approche multi-agent

Soutenu le : ^{03 octobre} ~~03 septembre~~ 2011, devant le jury composé de :

M ^{me} HEDJAZI	Promotrice
M ^{me} BENSETITI	Co-Promotrice
M ^{me} Benblichia	Présidente
M ^{elle} Aoussate	Examinatrice
M ^{elle} Arkid. L	Examinatrice

Dédicaces



A ma très chère mère Houria

Tu es l'exemple de dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

A mon très cher père Nourddine

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.

A mes frères et sœurs; Redhouane, Sihem, Samia, Riadh et Mohamed.

A mes beaux frères Faiçal et Smail.

Vous vous êtes dépensés pour moi sans compter. En reconnaissance de tous les sacrifices consentis par tous et chacun pour me permettre d'atteindre cette étape de ma vie.

A mon neveu, Fares meilleurs vœux de succès dans tes études.

A ma très chère nièce Malek.

A mon très cher neveu Wael.

Sans oublier Noor El Yakout et ma belle sœur Imène.

A mon binôme et mon fiancé Youcef qui a partagé avec moi les moments difficiles de ce travail.

A ma belle Famille.

A mes amis(es); Melyara, Halla, Nawel, Naziha, Youcef et Hamza.

A ma voisine Tata Nabila et sa fille Sihem.

Je vous remercie de votre patience, vous m'avez aidée toujours à avancer, vous êtes tous de grands(es) amis(es) si gentils(les), merci d'être toujours près de moi.

Hebib Warda

Dédicaces

A ma très chère mère Houria

Tu es l'exemple de dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

A mon très cher père Ahmed

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.

A mes frères et sœurs; Asma, Mohamed et Siryne.

Vous vous êtes dépensés pour moi sans compter. En reconnaissance de tous les sacrifices consentis par tous et chacun pour me permettre d'atteindre cette étape de ma vie.

A ma binôme et ma fiancée Warda qui a partagé avec moi les moments difficiles de ce travail.

A ma belle Famille.

Ghebriout Youcef.

Remerciements

Tout d'abord, nous tenons à rendre grâce à Dieu tout puissant pour nous avoir donné le courage et la détermination nécessaires pour finaliser ce travail et le mener à terme.

Nous adressons en premier lieu nos vifs remerciements à Mme HEDJAZI, chercheur au Centre de Recherche sur l'Information Scientifique et Technique (CERIST), pour nous avoir proposé un sujet de fin d'études riche et passionnant, et pour son encadrement et son suivi tout au long de l'exécution de notre PFE. Nous la remercions également pour son attention particulière, pour ses efforts et ses conseils dans la réalisation de notre PFE. Nos remerciements vont également à l'encontre du personnel du CERIST, pour nous avoir offert un terrain de stage pour concrétiser notre projet de fin d'études, et plus particulièrement à Mr MEZIANE (directeur du département systèmes d'information et systèmes multimédia du CERIST) pour son accueil, son orientation vers Mme HEDJAZI, et pour tous les services administratifs qu'ils nous ont rendus tout au long de notre stage au CERIST.

Nos remerciements vont ensuite à Mme BENSETTITI, enseignante à l'USDB au département d'informatique, pour son suivi tout au long de l'exécution de notre PFE et pour son attention.

Nos remerciements vont ensuite à tout l'ensemble de la Banque Centrale d'Algérie, plus particulièrement à Mr DAHIA NABIL pour son suivi tout au long de l'exécution de notre PFE, pour ses idées et son attention.

Nous remercions également l'honorable jury pour les efforts consentis à l'évaluation de notre travail.

Nous remercions également à Mr HEBIB DJAMEL pour son orientation vers Mr MEZIANE et Mme BENSETTITI.

Enfin, nos remerciements vont à l'encontre de nos amis et proches pour leur soutien moral et leur encouragement durant notre cursus universitaire.

Sommaire :

Introduction Générale

Partie I : ETAT DE L'ART

Chapitre I : LES SYSTEMES DE PAIEMENT DE MASSE

1. Introduction.....	5
2. Qu'est ce qu'un système de paiement ?.....	5
3. Qu'est ce qu'un système de télé-compensation?.....	7
4. Les caractéristiques du fonctionnement d'un système de télé-compensation....	7
5. Les risques générés par ces systèmes en cas de défaut d'un participant.....	7
5.1.La défaillance temporaire	8
5.2.Le défaut de règlement définitif.....	9
6. Les instruments de paiement et leur date de règlement.....	9
7. Les types des banques.....	11
8. Conclusion	11

Chapitre II : LES SYSTEMES MULTI-AGENTS

1. Introduction.....	13
2. Agent.....	14
2.1.Définitions.....	14
2.2.Architecture générale et fonctionnement d'un agent.....	15
2.3.Propriétés d'un agent.....	15
2.4.Typologie d'agents.....	16
2.4.1. Agent réactif	16
2.4.2. Agent cognitif.....	17
2.4.3. Agent hybride.....	17
2.5.Apprentissage des agents.....	18
2.5.1. Types d'apprentissage.....	18
2.5.1.1.Apprentissage symbolique.....	18
2.5.1.2.Apprentissage numérique.....	18

2.5.2. Paradigmes d'apprentissage	18
2.5.2.1.Apprentissage non-supervisé.....	18
2.5.2.2.Apprentissage supervisé.....	19
2.5.2.3.Apprentissage par renforcement.....	19
2.5.3. Techniques d'apprentissage.....	19
2.6.Applications des agents autonomes.....	19
3. Systèmes multi-agents (SMA).....	20
3.1.Définition.....	20
3.2.Interactions dans les SMA.....	21
3.2.1. Communication.....	22
3.2.1.1.Communication par partage d'information.....	22
3.2.1.2.Communication par envoi de messages.....	23
3.2.1.3.Les langages de communication dans les SMA.....	23
3.2.2. Cooperation.....	24
3.2.3. Négociation.....	25
3.3.Organisation d'un SMA.....	25
3.4.Méthodologies de conception des SMA.....	26
3.5.Plateformes de développement des SMA.....	28
3.6.Utilité et domaines d'application des SMA.....	29
4. Conclusion	30

Chapitre III : LES SYSTEMES DE CLASSEURS

1. Introduction.....	31
2. Notion de classeur	31
3. Représentation des données dans un classeur	31
4. Architecture générale d'un système de classeurs	32
5. Fonctionnement des algorithmes du LCS	33
5.1.Algorithme du Bucket Brigade	34

5.2.Création de nouvelles règles : covering.....	35
5.3.Algorithme génétique.....	36
6. Les différents types de systèmes de classeurs	37
7. Conclusion	38

Chapitre IV : QUELQUES MODELES D’OPTIMISATION DES ECHANGES
INTERBANCAIRES

1. Introduction.....	40
2. Optimisation bilatérale.....	40
2.1.Algorithme du Greedy.....	40
2.2.Algorithme LAS VEGAS GREEDY	42
2.3.L’algorithme Greedy++.....	43
3. Optimisation multilatérale.....	44
3.1.Modèle de Renault et Pecceu	44
3.2.Algorithme Multilatéral Greedy Las Vegas	45
4. Modélisation de la dynamique des systèmes de paiement	47
5. Modélisation du comportement d’un système bancaire par multi agent.....	50
6. Conclusion	51

Chapitre V : ETUDE DE L’EXISTANT

1. L’entreprise accueillante ‘CENTRE DE PRÉ-COMPENSATION INTERBANCAIRE’.....	52
2. Système d’échange et de compensation des paiements de masse.....	53
2.1.Vue technique du SCPM	53
2.2.Vue fonctionnelle du SCPM.....	54
2.3. Fichiers d’Entrées / Sorties	56
3. Présentation des flux des échanges entre les U.A.P. et le système A.T.C.I....	57
3.1.Flux de statistiques.....	57
3.2. Flux d’alarme.....	58
4. Organisation de la journée de compensation	58
5. Classification des moyens de paiement	58
6. classification des banques.....	59
7. Réforme du système de paiement les systèmes ATCI (Algérie Télé-Compensations Interbancaire)	60

8. Description des besoins.....	60
9. Conclusion.....	61

Partie II : Conception

Chapitre VI : Conception du système

1. Introduction.....	62
2. La méthode OMT	63
2.1.Introduction	63
2.2.Présentation générale de la méthode OMT.....	63
2.3.La démarche générale de développement.....	63
2.3.1. L'analyse	63
2.3.1.1.Modèle statique.....	64
2.3.1.2.Modèle dynamique.....	64
2.3.1.3.Modèle fonctionnel.....	64
2.3.2. Conception du système.....	65
2.3.3. Implémentation.....	65
3. Méthodologie de conception du SMA.....	65
4. Analyse du système	66
4.1.Spécification du système.....	66
4.2.Analyse des composants du système... ..	66
4.2.1. Identification des agents du système	66
4.2.1.1.Agent ATCI	66
4.2.1.2.Agents Banques.....	67
4.2.2. Schéma général du système.....	67
4.2.3. Environnement.....	68
4.2.4. Interactions	68
4.2.5. Organisation.....	69
5. Conception détaillée	69
5.1.Diagramme de cas d'utilisation	70
5.2.Agent ATCI	72
5.2.1. Architecture et environnement de l'agent ATCI	72
5.2.2. Classe d'Agent ATCI	73
5.2.3. Description de l'Agent ATCI.....	74
5.3.Agent Banque (AB).....	74

5.3.1. Module de raisonnement et d'apprentissage.....	75
5.3.1.1. Système de classeur	76
a. La structure du système de classeur de AB.....	76
b. Les paramètres et leur codification.....	77
c. Les Actions.....	82
d. Les valeurs des paramètres	82
e. Les poids des paramètres	83
f. Les cas possibles	85
g. Évaluation des règles: la fonction reward.....	85
5.3.2. Architecture d'un agent Banque	87
5.3.3. Classe d'Agent Banque	88
5.3.4. Description de l'Agent Banque.....	90
5.4. Processus global des traitements des opérations de paiement.....	90
5.5. Les diagrammes d'activités	93
5.5.1. Diagramme d'activité de l'ouverture d'une journée de compensation	93
5.5.2. Diagramme d'activité de la fermeture d'une journée de compensation.....	93
5.5.3. Diagramme d'activité du traitement appliqué sur une opération et prise de décision.....	94
5.5.4. Diagramme d'activité des traitements de l'agent banque.....	96
5.6. Les diagrammes d'interaction	98
5.6.1. Diagramme d'interaction « Ouverture de la journée de compensation »	98
5.6.2. Diagramme d'interaction « Fermeture de la journée de compensation »	99
5.6.3. Diagramme d'interaction « informer AB Remettante »	99
5.6.4. Diagramme d'interaction « informer AB Destinataire ».....	100
5.7. Conception de la Base de Données	101
5.7.1. Identification des classes de la BDD	101
5.7.2. Diagramme de classe de la BDD.....	101
5.7.3. Description des classes.....	102
6. Conclusion.....	102

Partie III : REALISATION DU SYSTEME

1. Introduction.....	103
2. Environnement de développement	103
2.1. Plateforme Multi-agent	103
2.2. Langage de programmation	104
2.3. SGBD.....	104
2.4. Module d'apprentissage	104
3. Algorithme d'exécution des Agents.....	105
3.1. Algorithme d'exécution de l'Agent ATCI.....	105
3.2. Algorithme d'exécution de l'agent banque	106
4. Architecture de l'application :.....	115
5. Présentation de l'application.....	116
6. Conclusion	130
Conclusion générale	

Liste des acronymes

ATCI : Algérie Télé-Compensation Interbancaire.

BBA: Bucket Brigade Algorithm.

CPI: Centre de Pré-compensation Interbancaire.

CS : Classifier System.

DAB : Distributeur Automatique de billets.

FIFO: First In First Out.

FIPA-ACL : Foundation for Intelligent Physical Agents – Agent Communication Language.

GAB : Guichet Automatique Bancaire.

JADE: Java Agent Development framework

LCS: Learning Classifier System.

RTGS: Real Time Gross Settlement.

SCPM: Système de Compensation des Paiements de Masse.

SIT : Système Interbancaire de Télé-compensation.

SMA : Systèmes Multi-Agents.

TPE : Terminal de Paiement Electronique.

UAP: Unique Access Point.

Les figures

<i>Fig. I.1: Système de paiement interbancaire et ses interfaces.</i>	6
<i>Fig. II.1. Architecture générale d'un agent.</i>	15
<i>Fig. II.2. Schéma d'un Système Multi-Agent.</i>	21
<i>Fig. II.3 Communication des agents par tableau noir.</i>	22
<i>Fig. II.4 Communication des agents par envoi de messages.</i>	23
<i>Fig. II.5 Organisation d'un système multi-agent.</i>	26
<i>Fig. II.6 Les différentes étapes de la méthodologie Voyelles.</i>	27
<i>Fig. III.1 Représentation d'un classeur.</i>	32
<i>Fig. III.2 Architecture générale d'un système de classeurs.</i>	33
<i>Fig. III.3 Fonctionnement des algorithmes du LCS.</i>	34
<i>Fig. IV.1 Réseau formé par les participants.</i>	45
<i>Fig. IV.2. Un modèle simple d'un système de paiement</i>	48
<i>Fig. IV. 3. Instruction et corrélation de paiement (a) et distribution Longueur de règlement de Cascade (b).</i>	49
<i>Fig. IV.4. Aperçu d'un modèle multi-agents d'apprentissage d'un système de paiement.</i>	50
<i>Fig. V.1: Diagramme basique du réseau du S.C.P.M.</i>	54
<i>Fig. V.2 Exemple des fichiers plats porteurs d'opérations en entrées / sorties.</i>	57
<i>Fig. V.3 Les séances de la journée de compensation.</i>	58
<i>Fig. VI.1 Schéma général du système.</i>	68
<i>Fig. VI. 2. L'organisation du système.</i>	69
<i>Fig. VI. 3 Diagramme de cas d'utilisation (UML).</i>	71
<i>Fig. VI.4 Architecture interne de l'agent ATCI.</i>	73
<i>Fig. VI.5 Représentation d'une règle du système de classeurs.</i>	77
<i>Fig. VI.6 Architecture interne de l'agent Banque.</i>	88
<i>Fig. VI.7 Enchaînement des traitements durant une journée de compensation (3 séances).</i>	92
<i>Fig. VI.8 Diagramme d'activité de l'ouverture d'une journée de compensation (UML).</i>	93
<i>Fig. VI.9 Diagramme d'activité de la fermeture d'une journée de compensation (UML).</i>	93

<i>Fig. VI.10 Diagramme d'activité du traitement appliqué sur une opération et prise de décision (UML).....</i>	<i>95</i>
<i>Fig. VI.11 Diagramme d'activité des traitements de l'agent banque (UML).....</i>	<i>97</i>
<i>Fig. VI.12 Diagramme d'interaction « Ouverture de la journée de compensation ».....</i>	<i>98</i>
<i>Fig. VI.13 Diagramme d'interaction « Fermeture de la journée de compensation ».....</i>	<i>99</i>
<i>Fig. VI.14 Diagramme d'interaction « informer AB Remettante ».....</i>	<i>100</i>
<i>Fig. VI.15 Diagramme d'interaction « informer AB Destinataire ».....</i>	<i>100</i>
<i>Fig. VI.16 Diagramme de classes de la base de données Locale (UML).....</i>	<i>101</i>
<i>Fig.VII.1 Interface d'authentification.....</i>	<i>116</i>
<i>Fig.VII.2 Interface Gestionnaire de tâches.....</i>	<i>118</i>
<i>Fig.VII.3 Interface Statistiques.....</i>	<i>118</i>
<i>Fig.VII.4 Interface Taux d'Evolution Annuel.....</i>	<i>119</i>
<i>Fig.VII.5 Interface Taux d'Evolution Mensuel.....</i>	<i>119</i>
<i>Fig.VII.6 Interface Taux de Rejet.....</i>	<i>120</i>
<i>Fig.VII.7 Interface d'Etat Précédent.....</i>	<i>120</i>
<i>Fig.VII.8 Interface de la Limite Banque.....</i>	<i>121</i>
<i>Fig.VII.9 Interface de calcul de la Ratio.....</i>	<i>121</i>
<i>Fig.VII.10 Interface de Contrôle.....</i>	<i>122</i>
<i>Fig.VII.11 Interface Catégorie Banque.....</i>	<i>122</i>
<i>Fig.VII.12 Interface Modification Catégorie Banque.....</i>	<i>122</i>
<i>Fig.VII.13 Interface Ajout.....</i>	<i>123</i>
<i>Fig.VII.14 Interface Modification Nom Banque.....</i>	<i>123</i>
<i>Fig.VII.15 Interface suppression d'une banque.....</i>	<i>123</i>
<i>Fig.VII.16 Interface Génération des Graphes.....</i>	<i>124</i>
<i>Fig.VII.17 Interface Taux des Opérations traitées en Nombre.....</i>	<i>124</i>
<i>Fig.VII.18 Graphe des soldes virtuels par banque avec le système actuel/journée..</i>	<i>125</i>
<i>Fig.VII.19 Graphe des Soldes virtuels des banques avec notre système/journée....</i>	<i>125</i>
<i>Fig.VII.20 Interface Taux des opérations traitées en nombre/journée.....</i>	<i>126</i>
<i>Fig.VII.21 Interface Evolution Annuelle des traitements des opérations.....</i>	<i>127</i>
<i>Fig.VII.22 Interface Evolution Mensuelle des traitements des opérations.....</i>	<i>127</i>

<i>Fig. VII.23 Interface Evolution du solde virtuel Avec Système Actuel.....</i>	<i>128</i>
<i>Fig. VII.24 Interface Evolution du solde virtuel Avec Notre Système.....</i>	<i>129</i>
<i>Fig.VII.4.26 Interface Génération du fichier XML.....</i>	<i>130</i>

Les tableaux

<i>Tab V.1 : Classification des banques</i>	59
<i>Tab VI.1 Représentation de la classe de l'agent ATCI</i>	74
<i>Tab VI.2 Codification binaire « Type Banque »</i>	77
<i>Tab VI.3 Codification binaire « Type Opération»</i>	78
<i>Tab VI.4 Codification binaire « Etat Banque »</i>	78
<i>Tab VI.5 Codification binaire « Limite Banque »</i>	79
<i>Tab VI.6 Codification binaire « Date de Règlement »</i>	79
<i>Tab VI.7 Codification binaire « Etat de la journée Précédente »</i>	80
<i>Tab VI.8 Codification binaire « Taux de Rejet »</i>	80
<i>Tab VI.9 Codification binaire « Taux d'Evolution par Mois »</i>	81
<i>Tab VI.10 Codification binaire « Taux d'Evolution par Année »</i>	81
<i>Tab VI.11 Codification binaire « Ratio »</i>	82
<i>Tab VI.12 Codification binaire « Classe Banque »</i>	82
<i>Tab VI.13 Codification binaire « ACTION»</i>	82
<i>Tab VI.14 Valeurs des paramètres</i>	83
<i>Tab VI.15 Les poids des paramètres « action Accepter »</i>	83
<i>Tab VI.16 Les poids des paramètres « action Retarder »</i>	84
<i>Tab VI.17. Les poids des paramètres « action Rejeter »</i>	84
<i>Tab VI.18 Les poids des paramètres « action Annuler »</i>	84
<i>Tab VI.19 Représentation de la classe de l'agent banque</i>	89
<i>Tab VI.20 Description des classes de la base de données</i>	103
<i>Tab VII.1 Description des composants de la fenêtre Gestionnaire de tâches</i>	117

Résumé

Un système de paiement de masse (ou système de compensation et de règlement net) est un système qui fonctionne le plus souvent en monnaie commerciale et traite des moyens de paiement de masse: virements, prélèvements, chèques, cartes... Ces systèmes assurent aussi des fonctions de compensation, qui permettent de réduire le nombre de règlements interbancaires à réaliser.

Le suivi, le contrôle et l'optimisation des échanges à travers des outils adaptés comme les systèmes multi-agents et les outils de l'intelligence artificielle s'avère utile à l'analyse quantitative du fonctionnement et des évolutions de ces systèmes ainsi que du comportement des participants (banque centrale, banques, Chèques postaux, etc.).

L'objectif du projet est l'ajout de composants logiciels au système existant pour l'étude et l'analyse des scénarios coopératifs entre les participants et de proposer des procédures d'optimisation multilatérale pour éviter les situations de blocage ou de contourner certaines défaillances du système. Ceci contribue à une meilleure efficacité du système.

Le projet consiste à concevoir et réaliser un système intelligent de suivi du système ATCI(Algérie Télé-Compensation Interbancaire) permettant de minimiser ses risques d'illiquidité par système multi-agent tel qu'à chaque participant (banque ou autre) est associé un agent adaptatif. La finalité est de proposer des scénarios d'optimisation du déroulement des règlements.

Mots clés

Système de paiement, procédure d'optimisation, système multi-agent, risque d'illiquidité, système de classeurs.

Abstract

A net settlement system (or clearing and net settlement) is a system that works most often deals with trade and currency of payment Mass transfers, withdrawals, checks, cards ... These systems also provide clearing functions, which reduce the number of interbank payments to be made.

Monitoring, control and optimization of trade through appropriate tools such as multi-agent systems and tools of artificial intelligence is useful in quantitative analysis of the functioning and evolution of these systems and the behavior of participants (central bank, banks, postal checks, etc.)..

The project is the addition of software components to the existing system for the study and analysis of scenarios cooperation between the participants and proposing multilateral optimization procedures to avoid deadlock situations or around certain failures system. This contributes to improved efficiency.

The project is to design and implement an intelligent monitoring system ATCI to minimize liquidity risk by multi-agent system as each participant (bank or other) is associated with an adaptive agent. The aim is to propose scenarios for optimizing the development of regulations.

Keywords

Payment system, process optimization, multi-agent system, liquidity risk, system files.

ملخص

نظام الدّفْع والذي يعرف باسم المقاصّة والتسوية الصّافية، هو نظام يعمل في أغلب الأوقات بالعملية التجارية ويعالج أنواعا خاصّة للدّفْع: التحويلات، العيّات، الصّكوك، البطاقات الإلكترونيّة. هذه الأنظمة تضمن أيضا كلّ خصائص المقاصّة، وذلك بتقليص عدد التسويات ما بين البنوك المراد تحقيقها. لضمان المتابعة، المراقبة، وتحسين التبادلات وجب استعمال وسائل متكيفة كنظام متعدّد الوكلاء ووسائل الذكاء الإصطناعي، التي تكون مفيدة للتّحليل الكمي لطريقة العمل والتّطور المستمر لهذه الأنظمة وسلوك المشاركين في نفس الوقت: البنك المركزي، البنوك الأخرى... الخ .

هدف المشروع الذي نصبو لتحقيقه، هو إضافة تركيبة (برنامج) للنظام الموجود حاليًا. وذلك قصد دراسة وتحليل كلّ السيناريوهات التّعاونية بين المشاركين في النظام (البنوك) واقتراح آليات تحسين متعدّدة الأطراف لتجنّب وضعيّات الحظر وتفادي الوقوع في أزمة ماليّة لبنك مشارك. كلّ هذا يساهم بشكل أو آخر في تحسين النظام المالي.

المشروع يتمحور حول تصميم نظام ذكيّ لمتابعة نظام المقاصّة الحالي، يسمح بتقليص أخطار الوقوع في ذانقات ماليّة بإشراك نظام (متعدّد الوكلاء)، بحيث أنّ كلّ مشارك (بنك) متعلّق بوكيل متكيف.

ختامًا نسعى لطرح سيناريوهات تحسين سير التسويات الماليّة.

الكلمات الأساسيّة: المقاصّة والتسوية الصّافية، نظام ذكيّ، نظام متعدّد الوكلاء، مخاطر السيولة.

Introduction générale

INTRODUCTION GENERALE

1. Contexte de travail

Un système de paiement de masse (ou système de compensation et de règlement net) est un système qui fonctionne le plus souvent en monnaie commerciale et traite des moyens de paiement de masse: virements, prélèvements, chèques, cartes... Ces systèmes assurent aussi des fonctions de compensation, qui permettent de réduire le nombre de règlements interbancaires à réaliser.

Avant l'année 2004, les travailleurs des différentes banques en Algérie ne disposaient pas d'un moyen automatique, qui facilite leurs tâches concernant les échanges interbancaires et leurs compensations et par conséquent, les décideurs avaient de la peine à se rencontrer à chaque fois qu'il s'agit d'une compensation des chèques, effets, virements, etc.

En 2004, la notion de la télé compensation est apparue, et qui veut dire : la compensation électronique. Cette nouveauté a enlevé l'obligation de se déplacer pour faire de la compensation. Cette solution réseau n'a pas très bien répondu aux attentes ; mais elle a quand même réduit le coût et désormais le temps consommé est devenu moins important. Cette préoccupation de ne pas avoir un mécanisme plus fiable et plus sûr a poussé les responsables économiques algériens à collaborer avec leurs confrères européens et américains qui sont plus en avance ; et en 2006 l'Algérie a vu la naissance de son premier système de compensation interbancaire, sous le nom : Algérie Télé Compensation Interbancaire (ATCI). Ce système appartient à la banque d'Algérie, qui a mis à sa disposition une unité autonome appelée CPI (Centre de Pré compensation Interbancaire), elle est indépendante et possède un équipage bien formé. De plus, L'ATCI est un système doté d'interfaces, qui sont attribuées à chaque participant (banque) pour le transfert des ordres de paiement. Les obligations de paiements sont divisées en deux parties: des paiements à gros montants et des paiements de masse. Pour cela on a deux systèmes qui traitent ces paiements: le premier est appelé RTGS (Real Time Gross Settlement) traitant les gros montants et en temps réel, c'est-à-dire ; les règlements de comptes se font directement après réception d'un ordre de paiement; contrairement au deuxième (ATCI) qui traite les paiements de masse

INTRODUCTION GENERALE

(chèques, prélèvements, cartes,... etc) mais en différé. À noter que les règlements des comptes de l'ATCI se font par le biais du RTGS, donc les deux systèmes forment une complémentarité. Notre étude se base sur le système ATCI. Ce système a fait preuve d'une stabilité et ses performances sont élevées; mais il reste toujours en évolution. Son rôle actuel se résume uniquement dans le routage des ordres de paiements, et pas au delà de ça. L'ATCI reste toujours ouvert à d'autres performances qui combleraient ses faiblesses.

Notre travail consiste à rajouter une couche au système qui joue le rôle d'aide à la décision pour minimiser les risques de manques de liquidités et garantir ainsi le suivi des soldes.

2. Problématique

Le système ATCI est le cœur des échanges interbancaires des banques algériennes. Toute opération émise d'une banque 'A' vers une banque 'B' doit traverser le système pour qu'elle arrive à destination et vice versa. Jusqu'à présent L'ATCI joue uniquement le rôle d'un routeur dans un réseau formé par les banques participantes; et cela l'empêche de retarder, rejeter; ou annuler une opération. Ces fonctionnalités restreintes lui permettent de router des ordres de paiement qui ont une influence négative sur la stabilité de l'économie du pays et même de causer la défaillance d'un participant (banque). La nécessité d'avoir une couche supplémentaire devient plus que jamais très importante. Son rôle est de superviser et analyser chaque opération reçue par le système. En sortie, l'opération peut être acceptée, retardée, rejetée, ou bien annulée. Pour réaliser cette tâche, notre couche doit être dotée d'un module intelligent qui prend en considération tous les facteurs qui interviennent à l'instant « T » comme: l'état de la banque (créditrice ou débitrice), le type d'instrument présenté dans l'opération, etc. Par conséquent le système donne un état à chaque opération reçue. A noter que le suivi et traitement des opérations se fait en temps réel, opération par opération.

INTRODUCTION GENERALE

En disposant d'un tel système la banque éloigne toute probabilité de tomber dans une situation de défaillance causée par une mauvaise gestion, et laisser la prise de décision au système ATCI, qui est désormais décideur.

3. Objectifs du travail

L'objectif du projet est l'ajout de composants logiciels au système existant pour l'étude et l'analyse des scénarios coopératifs entre les participants et de proposer des procédures d'optimisation multilatérale pour éviter les situations de blocage ou de contourner certaines défaillances du système. Ceci contribue à une meilleure efficacité du système.

Le projet consiste à concevoir et réaliser un système intelligent de suivi du système ATCI permettant de minimiser ses risques d'illiquidité par système multi-agent tel qu'à chaque participant (banque) est associé un agent adaptatif. La finalité est de proposer des scénarios d'optimisation du déroulement des règlements.

4. Organisation du mémoire

Afin d'atteindre les objectifs cités ci-dessus, notre mémoire s'articule autour de trois parties :

- **Etat de l'art:** contenant cinq chapitres:
 - **Chapitre I:** Les systèmes de paiement de masse, qui constitue une porte d'entrée et une mise en contexte global sur le monde des systèmes de paiements. Nous y abordons la définition d'un système de paiement, la définition d'un système de télé-compensation, les caractéristiques du fonctionnement d'un système de télé-compensation, les risques générés par ces systèmes, les instruments de paiements et les types de banques.
 - **Chapitre II:** Les Systèmes Multi-Agents (SMA), dans lequel nous présentons les agents, les systèmes multi-agents, les méthodologies de conception des SMA, les langages de communication et les plateformes de développement des SMA.
-

INTRODUCTION GENERALE

- **Chapitre III :** Les Systèmes de Classeurs, où nous présentons la notion de classeurs, la représentation des données dans un classeur, la représentation générale d'un système de classeurs, le fonctionnement des algorithmes du LCS (Learning Classifier System) et les différents types des systèmes de classeurs.
- **Chapitre IV :** Quelques modèles d'optimisation des échanges interbancaires, dans lequel nous présentons les travaux de recherche effectués dans le secteur d'optimisation des échanges interbancaires, l'optimisation bilatérale, l'optimisation multilatérale, la modélisation de la dynamique des systèmes de paiements et la modélisation du comportement d'un système bancaire par multi-agent.
- **Chapitre V :** Etude de l'existant, où nous présentons d'une manière générale le système de télé-compensation interbancaire algérien.
- **Conception :** contenant un seul chapitre:
 - **Chapitre VI:** Conception du système, dans lequel nous suivons la méthode de conception OMT pour la conception du système d'information et la méthodologie Voyelle pour la conception du SMA, nous décrivons aussi les différents diagrammes et l'architecture de notre système.
- **Réalisation:** contenant un chapitre.
 - **Chapitre VII:** implémentation et réalisation dans lequel nous présentons l'environnement de développement (langages et outils utilisés), les impressions d'écrans qui décrit les fonctionnalités de notre système.

En fin, la conclusion de ce mémoire synthétise nos principales contributions et donne quelques perspectives à notre travail.

1. Introduction

La qualité du système de paiement est un bon indicateur du fonctionnement d'une économie. Les progrès en matière d'intermédiation bancaire, le développement des institutions et la concurrence qui régit le marché bancaire ont permis l'évolution et la diversité des instruments et moyens de paiement. Les nouvelles technologies de transmission de données et les progrès en informatique sont à la base de l'amélioration des procédures de paiement et de recouvrement dans les pays développés et les pays émergents [TAI, 04].

2. Qu'est ce qu'un système de paiement ?

Un système de paiement est un ensemble d'outils techniques et technologiques obéissant à un cadre juridique, commercial et conventionnel qui interagissent dans une organisation d'échange connue ayant pour but de garantir le règlement financier des différents instruments de paiement utilisés dans la sphère économique réelle[BFR, 03].

Un système de paiements est composé :

- ❖ Des instruments de paiements.
- ❖ Des procédures de paiement et de recouvrement.

On distingue deux types de systèmes de paiements :

✚ Systèmes de paiements de gros montants (Real Time Gross Settlement RTGS)

Ces systèmes reposent principalement sur un fonctionnement en règlement brut, les opérations sont traitées individuellement, en temps réel, et en monnaie centrale.

Les systèmes de paiement de gros montant se caractérisent par un montant moyen des opérations très élevé.

✚ Systèmes de paiements de masse

Ces systèmes, qui fonctionnent le plus souvent en monnaie commerciale traitent des moyens de paiement de masse : virements, prélèvements, chèques, cartes... Ces systèmes assurent le plus souvent des fonctions de compensation,

qui permettent de réduire le nombre de règlements interbancaires à réaliser. Le plus souvent il s'agit d'une compensation multilatérale.

La différence primordiale entre le système ATCI et le système RTGS est que le règlement des opérations dans ce dernier se fait en temps réel, contrairement au système ATCI qui règle les opérations en fin de journée de compensation.

Le système RTGS traite des montants qui sont supérieurs à ceux du système ATCI.

La raison pour laquelle le système ATCI accumule les opérations est de laisser de la liquidité pour le système RTGS qui gère les gros montants.

À noter que L'imputation des comptes (règlement des comptes), se fait au niveau du RTGS.

Le schéma suivant présente le système de paiement interbancaire et ses interfaces :

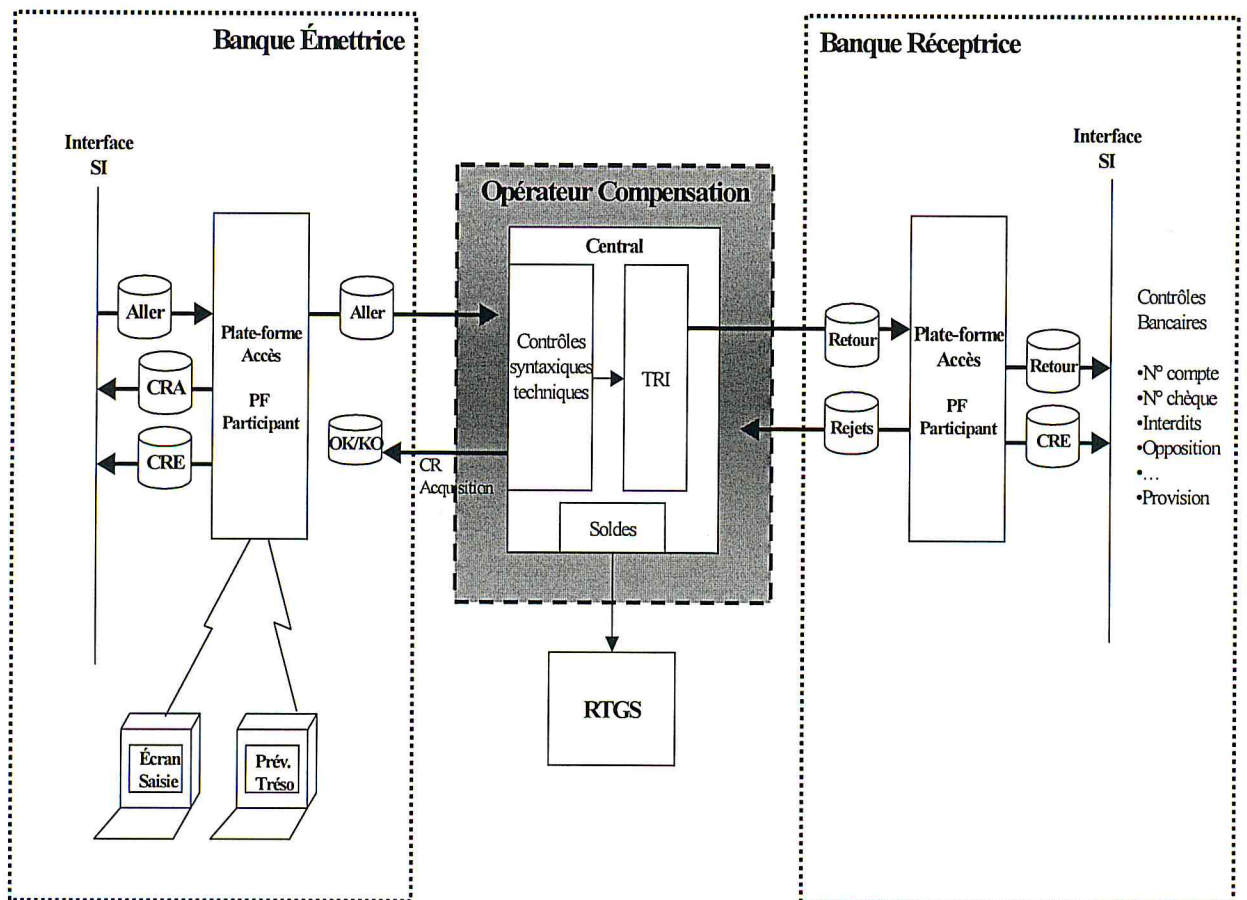


Fig. I.1: Système de paiement interbancaire et ses interfaces. [CHA, 04]

3. Qu'est ce qu'un système de télé-compensation?

Contrairement au système de compensation classique basé sur le déplacement physique des personnes et des valeurs, le système de la télé-compensation repose sur un système informatique d'échange de données numériques et d'images, limitant ainsi, au maximum les échanges physiques des données et des valeurs. Il s'agit tout simplement de la dématérialisation tant des supports que des échanges. L'opérateur du système est le CPI (Centre de Pré compensation Inter Bancaire)- Banque d'Algérie [TAI, 06].

4. Les caractéristiques du fonctionnement d'un système de télé-compensation

Le système de télé-compensation, présente deux caractéristiques essentielles auxquelles sont traditionnellement associés certains risques potentiels : ce sont des systèmes qui fonctionnent sur la base d'une compensation multilatérale des ordres de paiement et dont le règlement des obligations nettes des participants est différé par rapport au traitement des paiements unitaires sous-jacents [BFR, 03].

5. Les risques générés par ces systèmes en cas de défaut d'un participant

Les avantages et les risques inhérents à la compensation multilatérale associée à un règlement différé ont été analysés par les banques centrales. L'utilisation de ce mode d'extinction des obligations (retarder) a pour effet une réduction sensible des flux de paiements et des besoins de liquidité, comparée à d'autres modes d'extinction des obligations (compensation bilatérale et règlement sur une base brute). En contrepartie, la compensation multilatérale crée une interdépendance entre tous les participants au système considéré, puisque le défaut de règlement d'un seul participant en situation nette débitrice empêche le règlement des soldes de tous les autres participants - selon un principe de « tout ou rien » - et, par conséquent, de l'ensemble des opérations sous-jacentes traitées par le système. Cette situation crée pour les participants non défaillants des risques substantiels [BFR, 03].

➤ Risque en principal :

- ✦ Risque de liquidité;

⚡ Risque de coût de remplacement.

- **Risques de second rang** (c'est-à-dire sans incidence financière directe), qui peuvent être supportés par d'autres acteurs que les participants (gestionnaires des systèmes, banque centrale) comme par exemple entraîner un risque de réputation pour les gestionnaires du système et mettre en cause la confiance du public dans les moyens de paiement et dans la monnaie.

Cette typologie des risques inhérents aux systèmes nets à règlement différé permet d'identifier les risques associés à la participation dans le SIT (Système Interbancaire de Télé-compensation). Il convient, à cet égard, de distinguer entre les situations de défaillance temporaire et de défaillance définitive. Le risque en principal peut se définir comme le risque de perte de la totalité de la valeur d'un paiement, du fait d'un défaut de règlement [BFR, 03].

La procédure de révocation des échanges en cas de défaut de règlement (procédure consistant à recalculer les soldes nets des participants après exclusion des opérations du défaillant) devrait également éliminer le risque en principal, pour ce qui concerne le règlement interbancaire. Cependant, un risque en principal pourrait subsister si les participants non défaillants ne suspendaient pas également les paiements avec leurs clients, résultant d'opérations avec les clients du défaillant. Faute d'appliquer cette mesure, les non-défaillants risqueraient de payer leurs clients sans avoir reçu le règlement interbancaire correspondant. En revanche, les participants au SIT sont exposés aux autres types de risques propres aux systèmes de compensation multilatérale en cas de défaut : risques de liquidité, risque de coût de remplacement (le participant qui est à l'origine d'un défaut de paiement ; doit payer une somme d'argent, qui est considérée comme pénalité) et risques non financiers par exemples [BFR, 03]:

- ❖ Risques liés aux données (système d'information).
- ❖ Risques liés aux facteurs sociaux (conflits)
- ❖ Risques liés à la politique commerciale.

5.1. La défaillance temporaire

C'est la situation dans laquelle un participant devant couvrir un solde débiteur issu du SIT n'est pas à même de se conformer à ses obligations dans les délais impartis par les règles de la banque centrale, pour une raison ponctuelle

(problème technique de l'établissement, erreur humaine conduisant à une couverture tardive ou mauvaise anticipation de sa situation de liquidité).

Cette situation va se traduire par l'impossibilité d'imputation des soldes issus du système et au rejet des écritures par la Banque centrale vers le gestionnaire considéré. Cette situation est susceptible de perturber le bon fonctionnement, en raison des impacts sur sa chronologie et la gestion de trésorerie des établissements non défaillants qui vont subir un retard dans l'évolution programmée de leur liquidité. C'est pourquoi le défaillant se voit appliquer des pénalités financières et, en cas de récidive, pourrait se voir exclure du système par la Banque centrale [BFR, 03].

5. 2. Le défaut de règlement définitif

Ce défaut se caractérise par une situation où le défaillant ne serait pas à même, en raison d'une situation d'insolvabilité ou d'illiquidité durable, de couvrir son solde à bonne date de valeur. Il s'agit d'un scénario extrême qui, contrairement aux défauts temporaires, les conséquences seraient potentiellement considérables.

Le défaut de règlement définitif est susceptible de faire courir des risques de liquidité et de coût de remplacement substantiels, ainsi qu'un risque en principal résiduel (SIT), aux autres participants [BFR, 03].

6. Les instruments de paiement et leur date de règlement

Les instruments de paiement sont les instruments scripturaux définis par le comité de normalisation algérien [CHA, 04].

Il existe cinq types d'instruments de paiement, certains d'entre eux sont des instruments à débit d'autres à crédit. Le rôle de l'instrument change en fonction de sa position dans la banque.

6.1. Le chèque

Le chèque est un moyen de paiement délivré par les banques aux particuliers ou aux entreprises.

Le chèque est un moyen par lequel, le « tireur » (celui qui signe le chèque) donne l'ordre au « tiré » (une banque ou un autre organisme prévu par la loi) de

payer sur présentation un montant donné au « bénéficiaire ». C'est un moyen à crédit pour les banques remettantes.

On distingue trois types de chèque :

- Chèque électronique (pas de papier ni d'image).
- Chèque accompagné de son image scannée.
- Chèque accompagne de sa valeur papier.

Ils doivent être réglés au plus tard à **J+2**.

6.2. Les virements

Opération consistant à transférer une somme d'argent d'un compte bancaire ou postal à un autre compte. Le virement est un moyen à débit pour les banques remettantes.

Les différents types de virements :

- Virement standard de clientèle.
- Virement de banque à banque.
- Virement de régularisation.
- Virement en provenance de l'étranger.

Ils doivent être réglés au plus tard à **J**.

6.3. Les effets de commerce

Document émis par une personne ordonnant à une deuxième personne de payer une certaine somme d'argent à lui-même ou à une troisième personne.

Un effet de commerce peut être payable à vue, c'est-à-dire à la demande du bénéficiaire ; c'est notamment le cas des chèques. Il peut aussi être payable à une date déterminée : c'est le cas des lettres de change, des billets à ordre. Ils sont des moyens à crédit pour les banques remettantes.

On distingue deux types d'effets de commerce :

- Les lettres de change.
- Les billets à ordre.

Ils doivent être réglés au plus tard à **J+1**.

6.4. La carte

Les cartes permettent à son détenteur de créditer les achats de biens et de services qu'il effectue auprès de certains commerçants. Les cartes sont des moyens à crédit pour les banques remettantes.

Il existe deux types de cartes :

- Retrait sur DAB (Distributeur Automatique de billets)/GAB (Guichet Automatique Bancaire)
- Paiement sur TPE (Terminal de Paiement Electronique)

Ils doivent être réglés au plus tard à **J+1**.

6.5. Les prélèvements

Un prélèvement est une opération de débit (effectuée sur un compte).
Synonyme : retrait. Ils sont des moyens à crédit pour les banques remettantes.

Dans cette catégorie on ne voit qu'un seul type de moyens de paiements.

Ils doivent être réglés au plus tard à **J+1**.

7. Les types des banques

On distingue deux types de banques :

- ✚ **Les banques remettantes** ; ce sont les banques qui émettent les instruments de paiements.
- ✚ **Les banques destinataires** ; ce sont les banques qui reçoivent les instruments de paiements.

8. Conclusion

Dans ce chapitre, on a pu constater que la substitution des anciennes méthodes de règlement de paiement entre banque par un système de paiement à positiver tous le secteur économique. Les gens auparavant se rencontraient et échange leurs obligation de paiement, mais cette obligation de se déplacer à causer des problèmes de perte du temps, retardement de règlement, perte d'argent...etc.

Ce système de paiement adapté, qui a pour but d'automatiser les procédures de paiement, a joué un rôle très important dans la stabilité et le bon fonctionnement de l'économie ; et cela grâce à sa composition qui s'articule sur :

- ❖ Des instruments de paiements.
- ❖ Des procédures de paiement et de recouvrement.

Notre tâche se base sur l'étude de système ATCI qui a comme comportement de cumuler les procédures de paiement avant règlement en fin de journée, et cela par souci du manque de liquidité laissée au RTGS qui lui règle en temps réel.

Le système ATCI reste toujours performant, mais il joue uniquement le rôle d'un routeur qui fait passer les opérations d'une banque remettante à l'autre destinataire.

1. Introduction

Les systèmes multi-agents sont un domaine relativement jeune, apparu dans les années 80 (parmi les premiers travaux, on peut citer ceux de Victor Lesser et al. sur le système Hearsay [LEE, 80]), issu de la rencontre de l'Intelligence Artificielle et des Systèmes Distribués. Il y a plusieurs raisons à cela : d'une part, il y a l'idée que plutôt de tenter de résoudre un problème par une seule entité intelligente, il serait plus simple de faire coopérer plusieurs intelligences entre elles, afin de résoudre le problème par une équipe, ce qui simplifie la construction, la maintenance et l'exécution de ces systèmes complexes.

D'autre part, il existe certains problèmes qui sont distribués de manière inhérente, et que seule une résolution distribuée peut résoudre, par exemple la gestion décentralisée d'un réseau électrique ou la coopération de robots. Chaque entité a une perception ou des compétences limitées qui ne leur permettent pas individuellement de résoudre le problème. Ils utilisent la coopération et la coordination afin de résoudre ensemble le problème.

Finalement, de nombreux systèmes naturels procèdent d'un mécanisme distribué complexe. Ces systèmes peuvent être physiques (météorologie), biologiques (insectes sociaux [DRO, 93], épidémies), humains (mouvement de foule, économie) ou hybrides (exploitation des ressources naturelles [KOZ, 2000]). Ces systèmes sont souvent le siège de phénomènes émergents que l'on ne peut pas modéliser avec des outils standards. Afin de mieux les comprendre, il faut les simuler, en simulant le comportement d'un grand nombre d'entités atomiques afin de reproduire le phénomène émergent.

Tout ces travaux ont un point commun : ils visent à résoudre un problème ou à simuler un système en employant des entités (appelées agents) autonomes (ils satisfont leurs buts, les moyens de les satisfaire restant à leur discrétion), proactives (d'où le nom d'agent : ils agissent) et communicantes (que ce soit à travers un tableau noir, par messages ou par influences). Ce type de système est appelé système multi-agents [FER, 95].

2. Agent

2.1. Définitions

Dans la littérature des agents, il existe plusieurs définitions d'agent dont voici quelques unes :

- ✦ Selon Jacques Ferber [FER, 95], un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents.
- ✦ Selon Jennings, Sycara et Wooldridge [JEN, 98], un agent est un système informatique, **situé** dans un environnement, et qui agit d'une façon **autonome** et **flexible** pour atteindre les objectifs pour lesquels il a été conçu.

Les notions « **situé** », « **autonomie** » et « **flexible** » sont définies comme suit :

- **situé** : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement. Exemples : systèmes de contrôle de processus, systèmes embarqués, etc.
- **autonome** : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.
- **flexible** : l'agent dans ce cas est :
 - **capable de répondre à temps** : l'agent doit être capable de percevoir son environnement et élaborer une réponse dans les temps requis;
 - **proactif** : l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au "bon" moment;
 - **social** : l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de compléter ses tâches ou aider ces agents à accomplir les leurs.

2.2. Architecture générale et fonctionnement d'un agent

Lorsqu'un agent perçoit une situation dans l'environnement, il essaie de la reconnaître. Si la situation lui est familière, il peut enclencher un processus de planification afin de résoudre le problème. Il peut aussi reconnaître la situation en termes d'action et passer à l'exécution de la tâche (Reconnaissance- Exécution). Lorsque l'agent perçoit des situations qu'il connaît très bien, il peut faire intervenir son comportement réactif en passant directement à l'action (Perception-Exécution). S'il ne peut pas résoudre un problème (situation non-familière), il engage un processus de coopération pour demander de l'aide aux autres agents (Reconnaissance- Prise de décisions).

L'architecture suivante donne, de façon générale, la structure interne d'un agent :

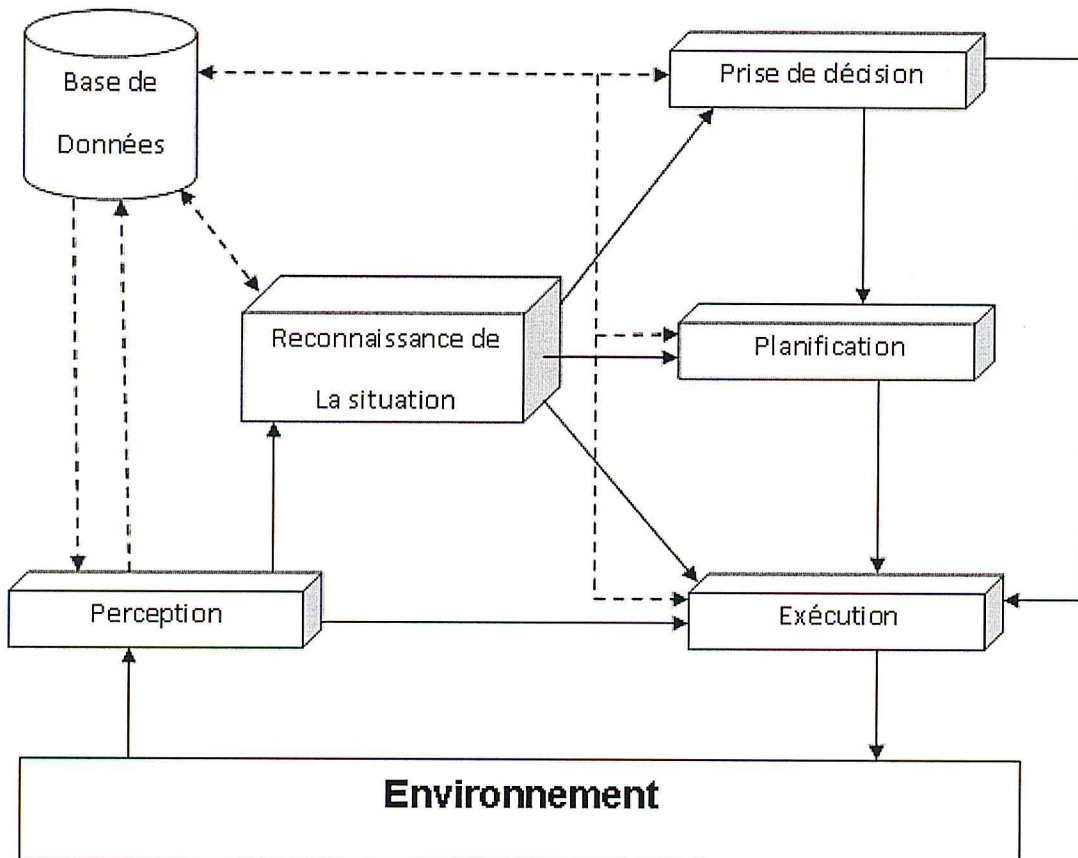


Fig. II.1. Architecture générale d'un agent [ARS, 01].

2.3. Propriétés d'un agent [JEN, 98]

De ces définitions d'agent ressortent les propriétés suivantes :

- **Autonomie** : un agent doit pouvoir agir seul, prendre une décision seul, en fonction d'informations lui provenant d'autres agents, de l'environnement. Il gère lui-même son état interne en fonction des informations qui lui parviennent.
- **Réactivité** : l'agent reçoit des informations de son environnement, et il doit être capable de réagir en conséquence.
- **Pro-activité** : un agent ne peut être autonome que s'il est capable de pro-activité c'est-à-dire qu'il peut avoir un comportement piloté par des buts en prenant l'initiative.
- **Efficacité** : c'est-à-dire la capacité à résoudre le problème, à atteindre ses buts.
- **Adaptation** : c'est la capacité qu'a un agent de résoudre de nouveaux problèmes à partir de son expérience. Pour ce faire, l'agent doit être capable d'apprentissage.
- **Robustesse** : c'est la propriété directement complémentaire à l'adaptation. C'est la capacité justement que l'agent n'évolue pas trop vite, sinon il va être influencé par de faux signaux. L'agent doit donc avoir un juste équilibre entre adaptation et robustesse.
- **Communication** : un agent est capable de communiquer avec d'autres agents (comportement social) dans un environnement multi agents.

2.4. Typologie d'agents [FER, 95]

D'après leurs propriétés, il existe trois grandes catégories d'agents :

2.4.1. Agent réactif

Les agents réactifs sont issus de la modélisation physique du monde réel : ils sont assimilés à des fonctions de transition. Ils possèdent les caractéristiques suivantes :

- Ils ne possèdent pas une représentation symbolique interne de leur environnement, si ce n'est que très partiellement.
- pas de mémoire de son histoire, ni de but explicite.
- Les actions consistent à réagir à des informations leur parvenant c'est-à-dire comportement de type stimuli/réponse.

2.4.2. Agent cognitif

Les agents cognitifs sont issus de la modélisation mentaliste du monde : ils sont assimilés à des systèmes experts. Ils possèdent les caractéristiques suivantes :

- ✦ Ils ont en général une représentation précise de leur environnement, et chaque agent possède des connaissances comprenant des informations, un savoir-faire.
- ✦ Ces agents sont dits intentionnels car ils ont des buts.
- ✦ Ils peuvent tenir compte de leurs passés pour résoudre de nouveaux problèmes.
- ✦ Ils doivent être intelligents. L'intelligence est définie le plus souvent en intelligence artificielle par la capacité d'apprentissage, et la capacité à s'adapter. Et l'agent intelligent doit faire preuve d'autonomie dans son intelligence. Il doit donc avoir des désirs et des intentions.

L'une des architectures les plus connues pour les agents cognitifs est l'architecture BDI. Dans le cadre du raisonnement pratique, un raisonnement orienté vers la prise en compte des états mentaux, les chercheurs ont développé l'architecture BDI de l'anglais *Belief, Desire, Intention* pour croyance, désir, et intention, une architecture bâtie autour du *raisonnement pratique*. Ces agents sont généralement représentés par un "état mental" ayant les attitudes mentales suivantes :

- *Les croyances* : Ce que l'agent connaît de son environnement.
- *Les désirs* : Les états possibles envers lesquels l'agent peut vouloir s'engager.
- *Les intentions* : Les états envers lesquels l'agent s'est engagé, et envers lesquels il a engagé des ressources.

2.4.3. Agent hybride

Pour la majorité des problèmes cependant, ni une architecture complètement réactive, ni une architecture complètement cognitive n'est appropriée. Comme pour les humains, les agents doivent pouvoir réagir très rapidement dans certaines situations (comportement réflexe), tandis que dans d'autres, ils doivent avoir un comportement plus réfléchi. Dans ce cas, une architecture conciliant à la fois des

aspects réactifs et cognitifs est requise. On parle alors d'architecture hybride, dans laquelle on retrouve généralement plusieurs couches logicielles.

2.5.Apprentissage des agents

2.5.1. Types d'apprentissage

On distingue deux types d'apprentissage artificiel, caractérisés par le type des données utilisées :

2.5.1.1.Apprentissage symbolique

L'apprentissage symbolique, comme son nom l'indique, manipule des symboles. Il fonctionne grâce à la mise en place de relations entre ces symboles par le biais de jugement. L'idée est donc d'élaborer des méthodes permettant d'extraire des connaissances structurelles ou décisionnelles à partir d'instances peu structurées. L'avantage principal de l'apprentissage symbolique est sa portée sémantique forte (par exemple un système à base de règles « si => alors » est compréhensible).

2.5.1.2.Apprentissage numérique

L'apprentissage numérique ne manipule pas de symboles, il traite uniquement des valeurs numériques quantitatives qui vont être manipulées afin de réaliser l'apprentissage. Les méthodes d'apprentissage numérique se révèlent être portables et permettent une grande adaptabilité, car non dépendant de symboles. Par contre, le fonctionnement interne du système est opaque. Il est très difficile de comprendre comment le système apprend (les réseaux de neurones artificiels en sont un bon exemple).

2.5.2. Paradigmes d'apprentissage

En apprentissage, le fait qu'intervienne (ou non) un enseignant afin d'aider l'apprentissage, en fonction des informations dont il dispose, définit le protocole d'apprentissage.

2.5.2.1.Apprentissage non-supervisé

En apprentissage non-supervisé, le système ne reçoit aucune information extérieure concernant les résultats attendus. La connaissance ou l'information est donc « incarnée » dans la structure des informations en entrée, et le système ou

l'algorithme doit découvrir lui-même les corrélations existantes entre les données qu'il a à traiter.

2.5.2.2. Apprentissage supervisé

Ici, un expert fournit la ou les solutions attendues, ou évalue l'erreur commise par le système. Durant l'apprentissage le système utilise les informations données par l'expert pour améliorer ses performances.

2.5.2.3. Apprentissage par renforcement

En apprentissage par renforcement, le système va essayer d'apprendre « seul » la façon d'améliorer ses résultats. La seule information dont il dispose est une *critique* sur son choix précédent (cette critique peut venir de l'environnement ou être une auto-évaluation). En fonction des *récompenses* ou *punitions*, il doit modifier ou non son comportement afin d'améliorer ses performances.

2.5.3. Techniques d'apprentissage

Parmi les nombreuses méthodes existantes d'apprentissage pour les agents, nous citons : les systèmes de classeurs, les algorithmes génétiques, les réseaux de neurones, etc. Dans notre projet, nous avons opté pour les systèmes de classeurs pour les raisons suivantes :

- Les systèmes de classeurs offrent des mécanismes d'évolution et d'apprentissage par renforcement grâce à des règles de types « condition => action » (apprentissage symbolique).
- Les systèmes de classeurs permettent de modéliser et de simuler le *comportement d'une entité* en fonction de *l'état de l'environnement* dans lequel elle évolue et de ses *états internes*.

Pour ces raisons évoquées, nous utiliserons donc les systèmes de classeurs pour réaliser le raisonnement et l'apprentissage de nos agents.

2.6. Applications des agents autonomes

La technologie agent est utilisée dans plusieurs applications, on retrouve :

- les agents interfaces : ce sont des entités qui assistent les utilisateurs dans l'exécution d'une tâche (exemple: le système de gestion du trafic aérien [KIN, 96]). Ils fournissent une assistance proactive à l'utilisateur pour une application précise.

- les agents mobiles [GRA, 98] : chargés de migrer d'une machine à une autre dans un réseau pour exécuter par exemple un code et ne pas surcharger le réseau.
- des agents émotionnels [HAY, 98] : ont été conçus en vue de refléter les "émotions" comme le feraient des humains placés dans la même situation. Ce type d'agent s'est déjà avéré utile dans des applications du type animation par exemple.

3. Systèmes multi-agents (SMA)

3.1. Définition

- **Définition 1**

D'après FERBER [FER, 95], on appelle système multi-agent (ou SMA), un système composé des éléments suivants:

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

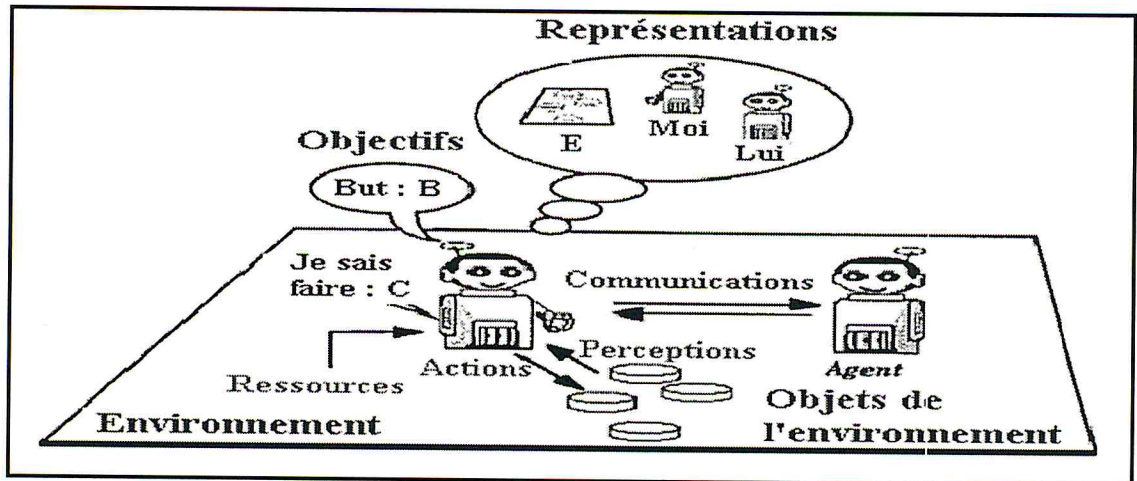


Fig. II.2. Schéma d'un Système Multi-Agent [FER, 95].

- **Définition 2**

Selon Jennings, Sycara et Wooldridge [JEN, 98], un système multi-agent se compose d'agents qui **interagissent** les uns avec les autres. Dans le cas le plus général, les agents agiront au nom des utilisateurs avec différents buts et motivations. Pour agir l'un sur l'autre avec succès, ils exigeront des capacités de **coopérer**, **coordonner**, et **négoier** les uns avec les autres, beaucoup comme le font les gens.

3.2. Interactions dans les SMA

Un système multi-agent (SMA) se distingue d'une collection d'agents indépendants par le fait que les agents interagissent en vue de réaliser conjointement une tâche ou atteindre conjointement un but particulier. Les agents peuvent interagir en communiquant directement entre eux ou par l'intermédiaire d'un autre agent ou en agissant sur leur environnement. Chaque agent peut être caractérisé par trois dimensions: ses buts, ses capacités à réaliser certaines tâches et les ressources dont il dispose. Les interactions des agents d'un SMA sont motivées par l'interdépendance des agents selon ces trois dimensions: leurs buts peuvent être compatibles ou non; les agents peuvent désirer des ressources que les autres possèdent; un agent X peut disposer d'une capacité nécessaire à un agent Y pour l'accomplissement d'un des plans d'action de Y.

Les types courants d'interaction incluent: la **communication** (échange d'informations entre agents); la **coopération** (travailler ensemble à la résolution d'un but commun) ; la **coordination** (organiser la résolution d'un problème de

telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées) ; la **négociation** (parvenir à un accord acceptable pour toutes les parties concernées).

3.2.1. Communication

Les agents communiquent entre eux à l'aide de protocoles de communication. Leur communication peut être de deux types : par *partage d'information* (communication indirecte) ou par *envoi de message* (communication directe).

3.2.1.1. Communication par partage d'information

En mode partage d'information, toute l'information sur le système est centralisée dans une structure de données globale. Les agents viennent lire et écrire dans cette base de données pour faire évoluer le système qui contient initialement les données du problème. Cette base de données, appelé « tableau noir » ou « blackboard ». D'apparence assez simple, ce mode communication centralisé implique que les agents effectuent un grand nombre de requêtes sur un seul et unique site central. De plus, le fonctionnement global du système n'est pas purement multi-agents puisque le comportement d'un des agents est dépendant du contenu d'une base de connaissances commune [DJA, 09].

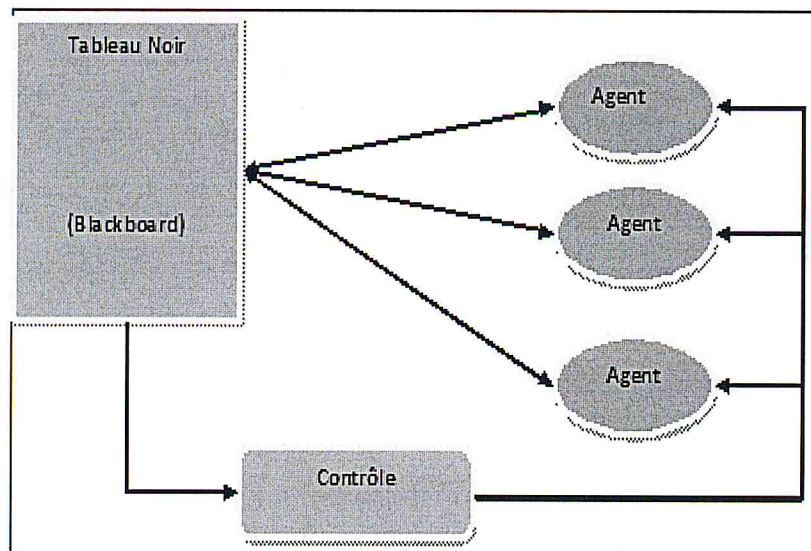


Fig. II.3 Communication des agents par tableau noir [DJA, 09].

3.2.1.2. Communication par envoi de messages

Les systèmes multi-agents fondés sur la communication par messages se caractérisent par le fait que chaque agent possède une représentation propre et locale de l'environnement qui l'entoure. Chaque agent va alors interroger les autres agents sur cet environnement ou leur envoyer des informations sur sa propre perception des choses. La communication se fait soit en mode point à point, soit en diffusion. Cette communication permet de réaliser un véritable système multi-agents puisque chaque agent possède sa propre base de connaissance. Par contre, il est plus difficile d'assurer une convergence globale du système. De plus, dans un environnement distribué, la communication entre un grand nombre d'agents peut très vite amener à une saturation du réseau compte tenu de la grande quantité de messages échangés [DJA, 09].

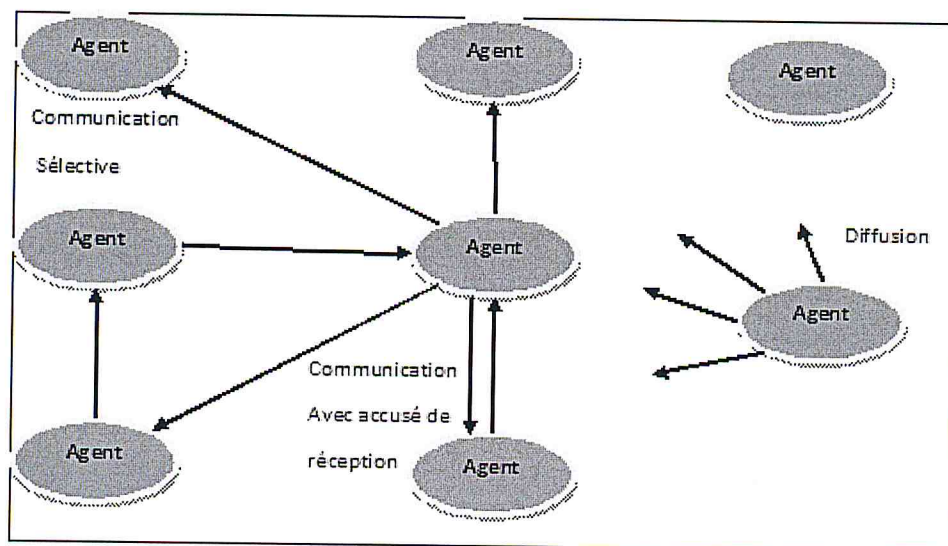


Fig. II.4 Communication des agents par envoi de messages [DJA, 09].

3.2.1.3. Les langages de communication dans les SMA [FER, 93]

Il existe plusieurs langage de communications nous allons définir quelques un.

a. KQML (Knowledge Query and Manipulation Language)

Il a été développé dans les années 90 au sein du groupe de travail « External Interfaces Group » du projet « Knowledge Sharing Effort » pour permettre la communication entre agents cognitifs. Ce langage se

base sur la théorie des actes de langages. KQML est un langage de haut niveau de communication et on peut considérer qu'il est composé de trois couches :

- Une couche de communication, qui décrit le niveau des paramètres de communication ;
- Une couche de message, qui contient un performatif et indique le protocole d'interprétation ;
- Une couche du contenu, qui contient des informations relatives au performatif soumis.

b. FIPA-ACL (Foundation for Intelligent Physical Agents – Agent Communication Language)

FIPA est une organisation à but non lucratif, regroupant des universitaires et des industriels qui tentent de mettre en place des standards dans le domaine des SMA. Comme KQML, FIPA-ACL est basé sur la théorie des actes de langages, il s'appuie sur la définition de deux ensembles :

- Un ensemble d'actes de communication primitifs, auquel s'ajoutent les autres actes de communication pouvant être obtenus par la composition de ces actes de base.
- Un ensemble de messages prédéfinis que tous les agents peuvent comprendre.

FIPA-ACL est superficiellement semblable à KQML. Sa syntaxe est identique à celle de KQML excepté différents noms pour quelques primitifs réservés. Ainsi, il maintient l'approche de KQML de distinguer le langage externe du langage interne. Le langage externe définit la signification prévue du message; le langage interne ou le contenu, dénote l'expression à laquelle s'appliquent les croyances, les désirs, et les intentions des interlocuteurs.

3.2.2. Coopération

La coopération entre les différents agents est un problème fondamental au sein des SMA. Les agents ont une connaissance plus ou moins précise des autres agents du système. Ils doivent pouvoir se représenter les compétences des autres agents, mais également les tâches que ces derniers sont en train de réaliser.

Les méthodes de répartition des tâches entre les différents agents est un problème majeur qui fait l'objet de nombreuses études, notamment dans le domaine du calcul réparti. Chaque problème doit être fractionné et réparti entre les différents agents qui produisent des solutions partielles qui doivent ensuite être fusionnées pour obtenir la solution globale.

Les modèles d'allocation des tâches peuvent être soit centralisés, un agent se charge alors de décomposer le problème et de le répartir, soit distribué, chaque agent est alors capable de décomposer le problème et de répartir les différentes tâches qui en résultent. Le modèle d'allocation distribué permet une meilleure utilisation des ressources des stations de travail mais la cohérence globale du système est plus compliquée à maîtriser [FER, 93].

3.2.3. Négociation

La collaboration entre agents peut parfois être de type conflictuel. Dans ce cas, les agents peuvent avoir des buts identiques mais des vues divergentes voire totalement opposées. Les conflits entre les différents agents peuvent être résolus de deux manières différentes :

- Grâce à un contrôle centralisé qui se charge, en dernier lieu, de prendre la décision.
- Par négociation entre les différents agents en conflits.

Si le principe de la négociation semble plus intéressant pour faire émerger des solutions innovantes, ce choix pose d'énormes difficultés notamment dues aux limitations linguistiques des agents et à une trop longue convergence de la négociation quand les conflits d'intérêts sont trop importants [FER, 93].

3.3. Organisation d'un SMA

L'organisation sociale d'un système multi-agent est la manière dont le groupe est constitué, à un instant donné, pour pouvoir fonctionner. Elle décrit l'ensemble des composants fonctionnels du système, leurs natures, leurs responsabilités et leurs besoins en ressources ainsi que les liens de communication entre les agents. Cette organisation peut être statique ou dynamique. Une société d'agents est constituée de trois éléments :

- un ensemble d'agents ;
- un ensemble de tâches à réaliser ;
- un ensemble d'objets associés à l'environnement [DUV, 01].

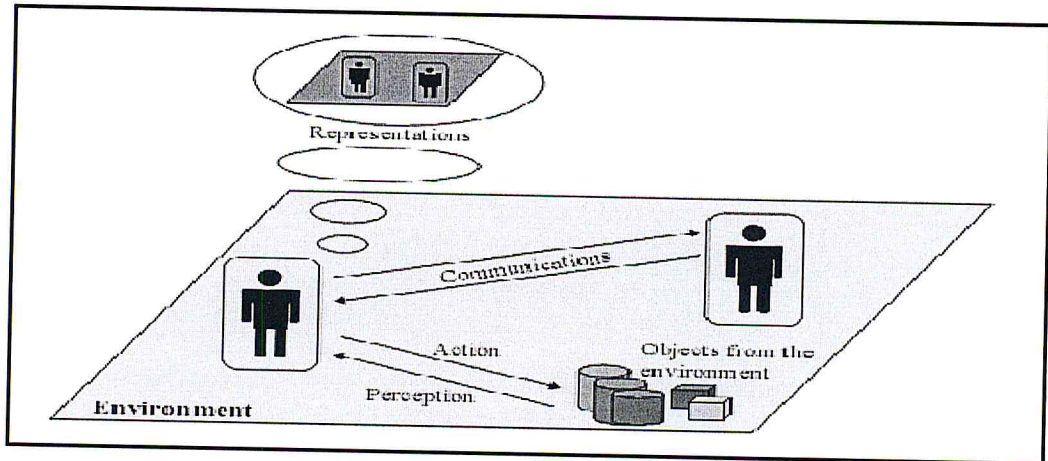


Fig. II.5 Organisation d'un système multi-agent [DUV, 01]

Un agent peut prendre la responsabilité d'effectuer une tâche s'il en a la capacité. Il prend alors un rôle dans le groupe. La réalisation d'une tâche suppose la manipulation d'objets de l'environnement. L'organisation peut être dynamique : le groupe se réorganise à chaque fois en fonction de la tâche à accomplir.

Il existe plusieurs modèles d'organisation, nous citons parmi eux :

- **l'organisation à membre unique**, où le seul agent présent dans l'organisation accomplit toutes les tâches.
- **l'organisation en groupe simple** : dès qu'un groupe existe, on peut avoir une coordination coopération afin d'atteindre un but commun.
- **l'organisation hiérarchique simple** : lorsqu'un agent ne peut pas réaliser une tâche complexe, cette dernière est divisée en sous tâches qui sont ensuite distribuées à un ensemble d'agents [DUV, 01].

3.4.Méthodologies de conception des SMA

Il existe plusieurs méthodologies de conception des systèmes multi-agents. Nous présentons dans ce qui suit une quand vas travailler avec :

3.4.1. Méthodologie Voyelles ou AIEO

Cette méthodologie proposée par Demazeau repose sur la décomposition d'un SMA en quatre parties, identifiées par les quatre voyelles : **A** pour Agents, **E** pour Environnements, **I** pour Interactions et **O** pour Organisations:

- **Agents** : concernent les modèles (ou les architectures) utilisés pour la partie active de l'agent.
- **Environnements** : sont les milieux dans lesquels sont plongés les agents, ils permettent de définir l'ensemble des capacités de perception et d'actions des agents.
- **Interactions** : concernent les infrastructures, les langages et les protocoles d'interactions entre agents, depuis de simples interactions physiques à des interactions langagières par actes de langage.
- **Organisations** : structurent les agents en groupes, hiérarchies, relations, etc.

En fin, il y existe également le composant **Utilisateurs (U)** qui représente l'ensemble des utilisateurs du système [DUM, 95].

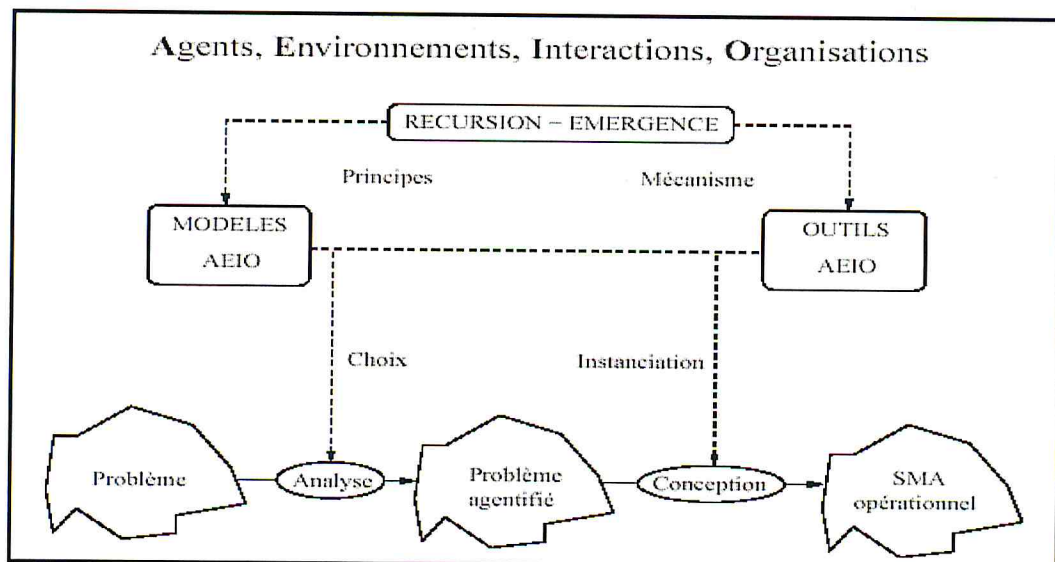


Fig. II.6 Les différentes étapes de la méthodologie Voyelles [DUM, 95]

D'après la méthodologie Voyelles, pour un problème à résoudre (ou un système à simuler) dans un domaine donné, l'utilisateur choisit le modèle d'agent, le modèle d'environnement, le modèle d'interaction et le modèle d'organisation qu'il instancie ensuite de manière à engendrer le système multi-agents opérationnel qui résoudra le problème dans le domaine considéré.

Il existe d'autre méthode comme :

- **Méthodologie GAIA** ; GAIA est le nom donné aux hypothèses formulées par l'écologiste James Lovelock.
- **Méthodologie AGR** ; Le modèle Agents/Groupes/Rôles a été conçu par Olivier Gutknecht et Jacques Ferber.

3.5. Plateformes de développement des SMA

Une plateforme de développement de systèmes multi-agent est un ensemble d'outils permettant la construction et la mise en service d'agents au sein d'un environnement spécifique. Ces outils peuvent servir également à l'analyse et au test du SMA ainsi créé ([HAC, 08]). Nous allons citer quelque plateforme :

3.5.1. La plateforme Jade (java agent development framework)

JADE est une plateforme multi-agent créé par le laboratoire TILAB, elle permet le développement de systèmes multi-agent et d'applications distribuées conformes aux normes FIPA (Foundation for Intelligent Physical Agents – Agent Communication Language); Aucune méthodologie n'est spécifiée, elle est entièrement écrite en java et fournit des classes qui implémentent JESS (Java Expert System Shell) pour la définition des comportements des agents, elle possède trois modules principaux (nécessaires aux normes FIPA) :

- ✦ DF « Director Facilitator » fournit un service de pages jaunes à la plate-forme.
- ✦ ACC « Agent Communication Chanel » gère la communication entre les agents.
- ✦ AMS « Agent Management System » supervise l'enregistrement des agents, leur authentification, leur accès et utilisation du système.

Les agents communiquent en utilisant le langage FIPA ACL (Agent Communication Langage).

Il existe d'autres plateformes comme :

- **La plateforme Madkit (multi-agent development kit)**

Développée en 1996 par Olivier GETKNECHT et Jaques FERBER au sein du Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) de l'Université Montpellier II.

▪ **La plateforme Zeus**

Zeus est une plateforme développée par British Telecom (Agent Research Programme of BT Intelligent Research Laboratory).

Nous nous intéressons à la plateforme Jade car nous allons travailler avec.

3.6. Utilité et domaines d'application des SMA

Les SMA sont à l'intersection de plusieurs domaines scientifiques : informatique répartie et génie logiciel, intelligence artificielle, vie artificielle. Ils s'inspirent également d'études issues d'autres disciplines connexes notamment la sociologie, la psychologie sociale, les sciences cognitives et d'autres. Les SMA possèdent également les avantages traditionnels de la résolution distribuée de problèmes notamment :

- **la modularité** : permet de rendre la programmation plus simple. Elle permet, de plus, aux systèmes multi-agents d'être facilement extensibles, parce qu'il est plus facile d'ajouter de nouveaux agents à un système multi-agent que d'ajouter de nouvelles capacités à un système monolithique.
- **la vitesse** : est principalement due au parallélisme, car plusieurs agents peuvent travailler en même temps pour la résolution d'un problème.
- **la fiabilité** : peut être également atteinte, dans la mesure où le contrôle et les responsabilités étant partagés entre les différents agents, le système peut tolérer la défaillance d'un ou de plusieurs agents. Si une seule entité contrôle tout, alors une seule défaillance de cette entité fera en sorte que tout le système tombera en panne.

Ils héritent aussi des bénéfices envisageables de l'Intelligence Artificielle comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité mais surtout, ils ont l'avantage de faire intervenir des schémas d'interaction sophistiqués.

D'après Ferber [FER, 95], nous pouvons considérer qu'il existe cinq grandes catégories d'applications des systèmes multi-agents:

- La résolution de problèmes : résolution des problèmes distribués et résolution distribuée des problèmes ;
- Construction de mondes hypothétiques ;

- La simulation multi-agent ;
- Robotique distribuée ;
- Conception kénétique¹ de programmes.

4. Conclusion

Les agents autonomes et les systèmes multi-agents représentent une nouvelle approche pour l'analyse, la conception et l'implémentation des systèmes informatiques complexes. La vision basée sur l'entité agent offre un puissant répertoire d'outils, de techniques, et de métaphores qui ont le potentiel d'améliorer considérablement les systèmes logiciels.

En accord avec Jennings [JEN, 98], nous pensons que la technologie agent va représenter durant les prochaines années, un nouveau paradigme de programmation pour le génie logiciel. Un nouveau paradigme "similaire" à la *programmation orientée objet* et qui d'ailleurs pourrait s'intituler *programmation orientée agent*.

Parmi les nombreuses applications des systèmes multi-agents, nous nous baserons sur les systèmes multi-agent pour réaliser les objectifs de notre projet.

¹ D'après Ferber, la kénétique a pour ambition de prôner un nouveau mode de conception de systèmes informatiques, cherchant à dépasser les techniques informatiques actuelles pour réaliser des logiciels distribués fonctionnant avec une grande souplesse et une grande adaptabilité à leur environnement [FER, 95]

1. Introduction

Les systèmes de classeurs sont des systèmes à base de règles de production qui permettent de résoudre des problèmes d'apprentissage par renforcement [GER, 01]. Le fonctionnement des LCS (Learning Classifier System) est basé sur deux principes : exploration et exploitation. L'exploration consiste à faire évoluer le système par la création de nouvelles règles plus adaptées et a pour objectif d'améliorer la capacité d'apprentissage d'un agent. L'exploration est réalisée par un algorithme génétique dont l'objectif est la création de nouveaux classeurs. L'exploitation, quant à elle, consiste à distribuer les récompenses renvoyées par l'environnement à la suite des actions exécutées par l'agent, ce qui améliore la valeur des règles ayant contribué à l'obtention de la récompense et diminue la valeur de celles qui en ont gène l'obtention. L'un des mécanismes de retribution le plus utilisé est « l'algorithme du bucket brigade ».

2. Notion de classeur

Un système de classeurs est constitué d'un ensemble de « **règles de production** », appelées « **classeurs** ». Chaque classeur est composé au moins d'une partie *condition* (qui correspond aux perceptions venant de l'environnement), d'une partie *action* (qui permet à l'agent d'agir sur son environnement) et d'une valeur *sélective* (qui aide à faire un choix entre plusieurs règles). Une règle est équivalente à :

si (C = S) alors (A) où C est la condition, S la situation et A l'action à effectuer

3. Représentation des données dans un classeur

L'alphabet de la condition est ternaire. En plus des deux bits de l'alphabet binaire (0 et 1), on introduit le caractère « # » qui symbolise le fait que la valeur de la perception n'entre pas en compte dans la décision. Par exemple, la condition représentée en ternaire par 1#01 correspond à deux situations différentes dans l'environnement : 1001 et 1101.

- condition {0, 1, #}.
- action {0, 1}.
- force (nombre entier ou réel).

Condition	Action	force
[01#01#01]	[0111000101]	69

Fig.III.1 Représentation d'un classeur.

4. Architecture générale d'un système de classeurs

Un système de classeurs est constitué de quatre modules :

- **l'interface d'entrée** : permet de traduire la perception externe en messages formatés à destination de la liste des messages.
- **l'interface de sortie** : permet aux messages formatés de la base des règles d'être traduit en action réelle sur l'environnement.
- **la liste des messages** : il existe trois types de messages.
 - **Les messages d'entrées** : ces messages sont construits par l'interface d'entrées. Ces messages peuvent s'apparier avec la partie condition de certains classeurs de la base de règles qui émettront alors un message à destination de la liste des messages.
 - **Les messages internes** : ces messages sont émis par la base de règles. Ils peuvent déclencher d'autres classeurs de la base. On peut ainsi déclencher des suites de règles.
 - **Les messages de sorties** : ces messages sont émis par les classeurs de la base des règles.
- **La liste de classeurs** : elle contient les règles (les classeurs). Un classeur est déclenché si un message de la liste des messages correspond à sa partie condition. Un message correspond à une situation si et seulement si chaque caractère de sa partie condition est égal au caractère de la situation ou à « # ».

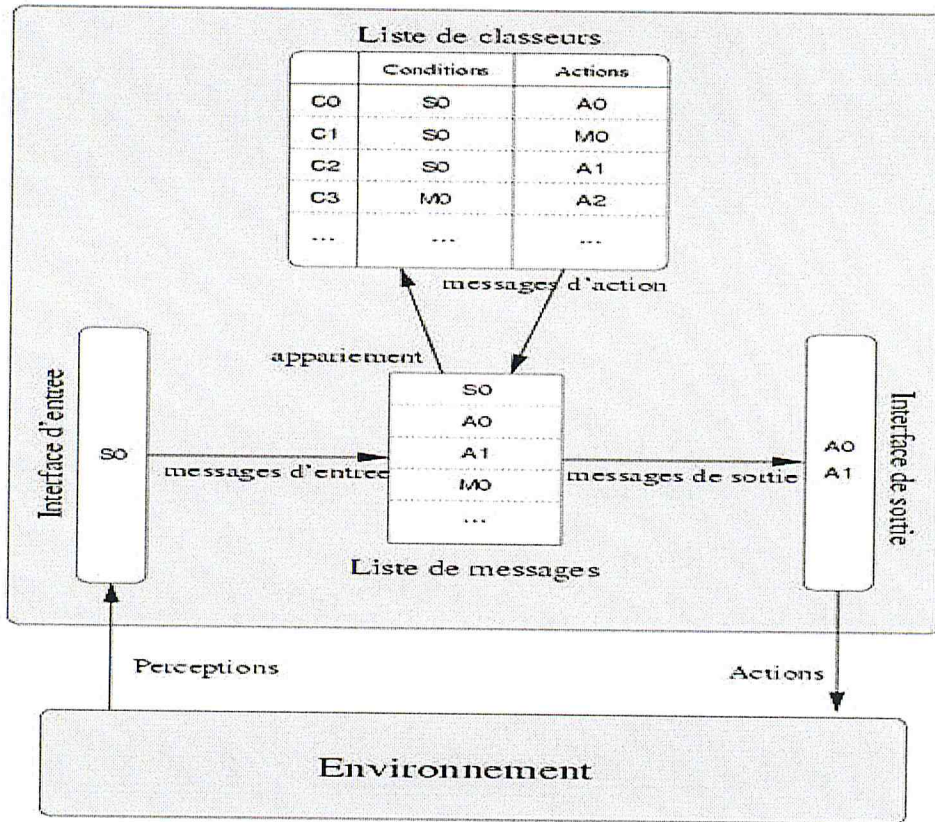


Fig.III.2 Architecture générale d'un système de classeurs [ERW, 03]

Lorsque l'agent perçoit une situation de l'environnement, il recherche dans son ensemble de règles si une ou plusieurs conditions s'apparient avec cette situation. S'il en trouve, il choisit de réaliser l'action qui correspond au classeur qui possède la plus forte valeur sélective (celle qui lui apportera potentiellement la plus grande récompense). Sinon, une nouvelle règle est créée pour cette situation et on réalise une action au hasard. La valeur sélective représente la capacité du classeur à résoudre un problème.

5. Fonctionnement des algorithmes du LCS

Un LCS est un système adaptatif qui apprend à effectuer la meilleure action compte tenu de son entrée. Par "meilleures" on entend généralement l'action qui va recevoir la récompense ou la plupart des renforts de l'environnement du système. Par "entrée", on entend l'environnement comme détecté par le système, généralement un vecteur de valeurs numériques. L'ensemble des actions disponibles dépendent du contexte de la décision, par exemple d'ordre financier, les actions pourraient être "Accepté",

"Retardé ", etc. En général, une LCS est un modèle simple d'un agent intelligent d'interagir avec un environnement.

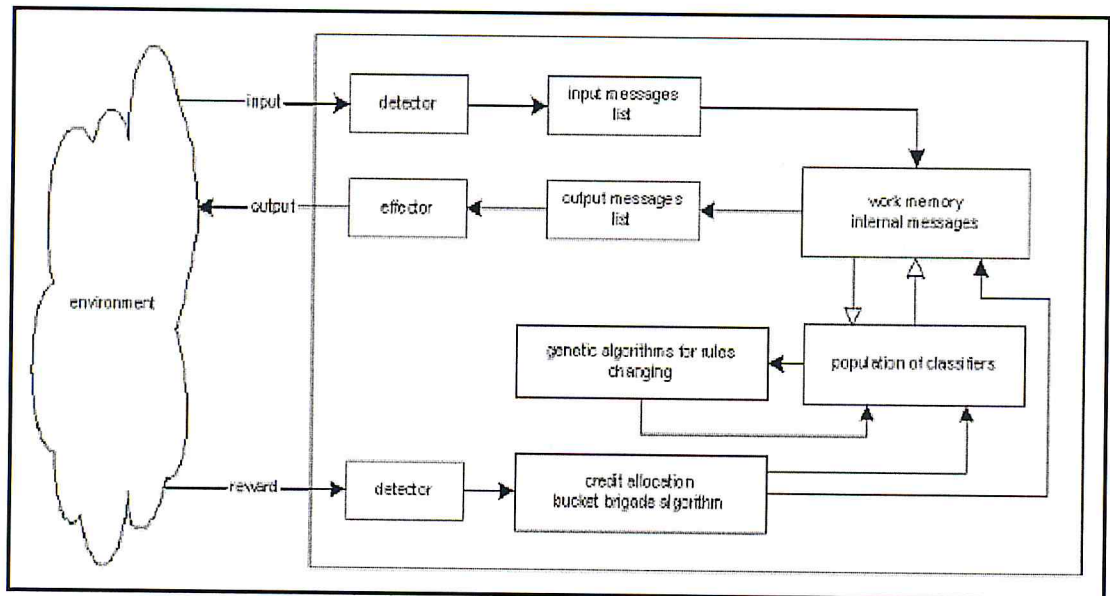


Fig.III.3 Fonctionnement des algorithmes du LCS [THE, 04]

5.1. Algorithme du Bucket Brigade

L'algorithme du bucket-brigade (BBA) [JOH, 87] permet d'anticiper le renforcement que recevra chaque classeur lorsque l'action déclenchée en fin de cycle aura été évaluée et aura reçu le renforcement correspondant de la part de l'environnement. Pour décrire l'algorithme, Holland utilise une analogie avec les intermédiaires en économie : ces personnes touchent une commission sur les affaires qu'ils apportent et paient une commission à leurs propres intermédiaires. En effet, le BBA fonctionne ainsi : chaque classeur paie une offre (**Bid**) immédiat aux classeurs qui lui ont permis de se déclencher (il s'agit des classeurs dont le message émis dans la liste s'appartient à la partie condition du classeur considéré ou alors un classeur qui a déclenché l'action permettant de se retrouver dans l'état actuel).

Lorsque plusieurs classeurs sont élus, il faut les répartir. Pour cela, chacune des règles déclenchées participe aux enchères (**auktion**). Chaque classeur fait donc une offre (Bid) que l'on calcule de la manière suivante :

$$\text{Bid} = \text{Force} * C_{\text{bid}} * \text{Spécialisation}$$

Où C_{bid} est une constante comprise entre 0.1 et 0.2. La spécialisation est le nombre de symboles # de la partie condition d'une règle divisée par sa longueur, elle est définie comme suit : **Spécialisation** = $(L-W+1)/(l+1)$.

Où L est la longueur de la partie condition de la règle, W est le nombre de symbole « # » dans la condition. La constante 1 est utilisé pour permettre aux règles constituées entièrement de « # » de participer quand même. Moins une règle contient de symboles #, plus elle est spécialisée.

Ensuite, il suffirait de choisir celui qui fait la meilleure offre (**Bid**). Mais il convient plutôt de laisser un peu de place au hasard. Alors on ajoute à l'offre une variable aléatoire $N(\sigma_{bid})$ où N est une fonction de l'écart-type des enchères σ_{bid} . Puis on choisit la plus grande valeur.

Le classeur nouvellement élu doit maintenant s'acquitter d'une taxe (c'est le « pari » qu'il fait sur la qualité de l'action qu'il déclenche). L'offre que le classeur fait est retirée de son potentiel de jeu (ici sa force) et est donné soit à l'environnement (si la situation venait de l'environnement) soit au classeur précédemment déclenché (si la situation correspondait en fait à un message interne). Ensuite, le classeur récupère sa mise (ou une partie seulement) soit à partir d'un autre classeur qu'il aurait déclenché, soit de l'environnement (si l'action déclenchée donne un résultat intéressant).

5.2.Création de nouvelles règles : covering [JOH, 87]

Il paraît évident que la base de règles ne possède pas des classeurs correspondant à toutes les situations rencontrées. Il faut alors utiliser un mécanisme, appelé « **covering** », qui permet au système de classeurs, lorsqu'il rencontre une situation inconnue, de s'y adapter. Pour cela, on génère un nouveau classeur constitué dans sa partie condition de la situation courante où certains bits ont été remplacés par le symbole « # » avec une probabilité P . (P est une valeur choisi de façon empirique, en général égale à 0.33). On génère ensuite une action quelconque au hasard. La force doit correspondre à la moyenne des forces de la base des règles.

Il faut ici mettre en avant le fait que la création de nouvelles règles du fait de la rencontre d'une situation inconnue est une chose qui ne doit pas se produire

souvent au risque de rendre l'apprentissage impossible. Le renforcement des règles conduisant à une action productive permet au système de conserver les règles les plus intéressantes mais ne constitue pas en soi un apprentissage. Pour faire cela, on utilise un algorithme génétique sur l'ensemble des règles du système de classeurs.

5.3. Algorithme génétique

Les systèmes de classeurs possèdent un mécanisme pour faire évoluer une population de classeurs : chaque classeur est évalué tout au long de l'interaction de l'agent avec l'environnement. Dans la plupart des systèmes de classeurs, la création et la sélection de classeurs fiables est réalisée par des algorithmes génétiques. Un algorithme génétique est appliqué à une population de classeurs (ou une fraction de celle-ci) et se déroule en générations successives. A chaque génération, des opérateurs de croisement et de mutation (présentés ci-dessous) permettent de créer de nouveaux individus à partir des meilleurs individus de la population, et un mécanisme de sélection supprime les moins bons. Ainsi, au fur et à mesure des générations, les individus sont de plus en plus adaptés à la résolution du problème.

Le processus d'un algorithme génétique est généralement comme suit :

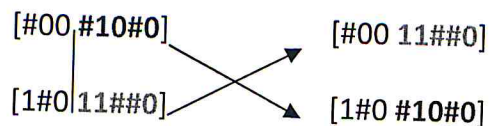
- ✓ Sélection d'un ou plusieurs individus *parents* dans la population.
- ✓ Création de nouveaux individus *filis* par copie directe des parents (ou clonage).
- ✓ Application de croisements et/ou de mutations sur ces nouveaux individus avec des probabilités fixes.
- ✓ Mise à jour de divers paramètres (valeur sélective. . .).
- ✓ Insertion de ces nouveaux individus dans la population.
- ✓ Suppression d'un certain nombre d'individus considérés comme inadaptés à la résolution du problème.

Les différents opérateurs utilisés par l'algorithme génétique pour la création de nouveaux individus sont :

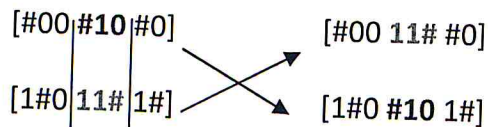
Crossover

Le crossover consiste, pour deux conditions, à s'échanger une partie de leurs attributs par croisement. Il existe plusieurs techniques pour le faire, les deux principales étant :

- *le one-point crossover* ; Consiste à échanger tous les attributs à partir d'un certain index choisi aléatoirement :



- *le two-point crossover* ; Consiste à échanger tous les attributs contenus entre deux index choisis aléatoirement :



6. Les différents types de systèmes de classeurs

De nombreux systèmes de classeurs apprenants sont apparus LCS (basé sur la récompense [JOH, 87]), ZCS (basé sur la force [WIL,94]), XCS (basé sur la précision [WIL,95]), ACS (basé sur l'anticipation) [STO,98]), etc.). Ces systèmes sont présentés brièvement dans les sections suivantes.

6.1. Le LCS de Holland

Le système de base de Holland LCS utilise une quantité nommée « force » pour évaluer les classeurs. Cette « force » est obtenue par un algorithme « Bucket brigade » qui utilise la rétro-propagation de la récompense environnementale sur les classeurs. Elle constitue une valeur sélective pour les algorithmes génétiques.

Ce système est également caractérisé par une liste de messages. Cette liste inclut les messages traduisant la perception de l'environnement et ceux traduisant les actions ou les messages de sorties. Si la condition d'un classeur est appariée à un message de la liste, le classeur est activé et les messages qui ont activé ce classeur sont ultérieurement éliminés. Le classeur envoie ensuite sa partie action à la liste de messages. Les messages qui peuvent être interprétés sont transformés en action et sont ensuite supprimés. D'autres messages vont rester dans la base et

apprentissage artificiel en projetant de réaliser un objectif à court terme et ne pas se baser sur l'anticipation des actions des autres dans un objectif à long terme comme c'est le cas des ACS. L'apprentissage réalisé par les LCS se base sur l'amélioration de la récompense immédiate des règles de production et consiste à maintenir les règles qui en produisent plus en supprimant celles qui en produisent peu, ce qui fait qu'il s'approche de notre cadre d'étude.

Pour réaliser l'apprentissage et l'adaptation des agents dans notre projet, nous nous baserons sur les systèmes de classeurs.

1. Introduction

La loi FIFO est appliquée par tous les systèmes de paiement ; mais sa stratégie de règlement devient un obstacle pour le règlement des paiements qui sont entrés plus tard à la file d'attente et qui présentent des montants très important. La loi FIFO ignore l'importance d'un paiement par rapport à un autre.

Le but de ces modèles est d'investiguer autres types d'algorithmes qui ne suivent pas nécessairement la loi FIFO pour le règlement des opérations, et de quantifier à quelle degré cette optimisation peut être plus efficace par rapport aux algorithmes FIFO.

2. Optimisation bilatérale

L'optimisation bilatérale consiste à examiner deux participants A et B et de considérer uniquement les paiements de A à B et de B à A. L'objectif de l'optimisation bilatérale est de régler simultanément un ensemble de paiements enfilés pour un maximum de valeur possible. A noter que le nombre de paiement à régler reste aussi important mais il est considéré comme un second objectif.

2.1. Algorithme du Greedy [GÜN, 98]

Güntzer et al ont proposé en 1998 d'utiliser les algorithmes Greedy comme outils, qui visent dans leurs comportements d'optimiser la valeur du phénomène étudié (dans notre cas le volume de compensation).

Cette étude qui a comme objectif de favoriser le volume de compensation, mène à optimiser le système de compensation allemand appelé EAF-2, pour réduire l'effet d'insolvabilité des participants et éviter ainsi une chaîne de défaillance provoquée par un participant qui a un déficit.

Les paiements ne sont pas conservés selon leur ordre d'arrivée, mais en fonction de leur valeur.

L'algorithme Greedy essaie d'abord de régler tous les paiements en file d'attente entre les deux banques simultanément.

Si cela n'est pas possible alors:

Tous les paiements d'un participant qui n'ont pas assez de liquidité sont désactivés, et sont ensuite réactivés à chaque fois que cela est possible, compte tenu de la liquidité contrainte dans l'ordre décroissant de leur valeur. Ce processus est répété jusqu'à ce que tous les paiements soient désactivés ou jusqu'à ce qu'une

solution a été trouvée.
Les grands paiements sont plus favorisés au détriment des paiements qui sont entrés dans le système au début.

Un des avantages de l'algorithme Greedy, c'est que les files d'attente ne seront pas bloquées en raison d'un seul paiement de très grande valeur qui empêche tous les paiements ultérieurs d'être réglés. Une propriété très intéressante de l'algorithme Greedy est qu'il donne une solution qui maximise la valeur des paiements réglés lorsque l'ordre des valeurs des paiements en file d'attente est (superincreasing : $a_i \geq \sum_{k=i+1}^N a_k$ tel que a_i est un paiement de la banque A vers B).

Cet algorithme est optimal lorsqu'il traite un petit nombre de paiements.

L'algorithme peut être performant même si l'ordre n'est pas (superincreasing), mais avec aucune garantie.

L'algorithme est simple et rapide.

Le temps dont on a besoin pour ordonner n paiements est de $N \cdot \log(N)$; mais une telle tâche ne doit être effectuée qu'une seule fois [GÜN, 98].

- **Détail de l'algorithme**

Activer tous les paiements entre les deux banques considérées.

Quand l'une des banques a un fond virtuel négatif.

Début :

Désactiver tous les paiements de la banque défaillante (supposons que ce soit la banque A).

Prendre les paiements de la banque A (a_i) $i=1 \dots N$ du plus grand a_1 au plus petit a_N .

Un paiement a_k peut être :

Si $a_k >$ fond virtuel de la banque A.

Alors ce paiement ne peut pas être réactivé.

Si non

Activer a_k .

Passer au prochain paiement.

Fin.

Régler tous les paiements activés.

L'algorithme greedy fournit une solution optimale quand les paiements sont dans un ordre superincreasing, mais autrement, la garantie d'obtention d'une bonne solution est éloignée.

2.2. Algorithme LAS VEGAS GREEDY :

Renault et Pecceu ont étudié en 2007 les effets des règles d'enfilement et les algorithmes d'optimisation. Renault et Pecceu ont essayé d'introduire des améliorations concernant l'algorithme Greedy.

La première idée c'est d'introduire une certaine flexibilité à cet algorithme, qui à chaque fois essaye de régler les ordres de paiement selon l'ordre décroissant.

L'autre idée c'est d'essayer de régler un maximum de possibilité de paiement.

Deux algorithmes sont introduits pour régler ce genre de problème :

L'idée est d'introduire une flexibilité à l'algorithme Greedy qui active les paiements selon l'ordre décroissant de leurs valeurs. Prenons par exemple : deux banque A et B tel que (a_1, \dots, a_n) .

Sont les paiements de A vers B. il est clairement optimal d'activer le paiement a_i si la séquence suit l'ordre superincreasing. si non, il n'est pas clair de l'activer ou pas. Donc cet algorithme essaye de combiner les deux solutions [HAR, 07].

- **Détail de l'algorithme**

Activez tous les paiements entre les deux banques considérées.

Quand l'une des banques a un fond virtuel négatif.

Début :

On désactive tous les paiements de la banque défaillante (on suppose que ce soit la banque A).

On prend les paiements de la banque A (a_i) $i=1\dots N$ de plus grand a_1 au plus petit a_N .

Un paiement a_k peut être :

Si $a_k >$ fond virtuel de la banque A.

Alors ce paiement ne peut pas être réactivé.

Sinon

Calculer : $R_k = \sum_{i=k+1}^N a_i$ La somme des paiements les plus que a.

Si $R_k <$ au fond virtuel de la banque A

Alors a_k est activé.

Si non :

Soit : $p_k = \min\left(\frac{a_k}{R_k}, 1\right)$.

Réactiver le paiement a_k avec une probabilité de p_k .

On passe au prochain paiement.

Fin.

Régler tous les paiements activés.

2.3.L'algorithme Greedy++ « Renault et Pecceu » [HAR, 07]

Cet algorithme essaie de régler un maximum de possibilités de paiements.

- **Détail de l'algorithme:**

Activer tous les paiements entre les deux banques considérées.

Quand l'une des banques a un fond virtuel négatif.

Début :

Soit : $G = \sum_i a_i x_i$ et $H = \sum_i b_i y_i$

L'erreur est définie comme suit : $\Delta = |G - H - \frac{1}{2}(S_A - S_B)|$.

On prend 10 paiements (de A ou de B) les plus proche de l'erreur (on prend les a_i et b_i qui minimise $|\log(\frac{a_i}{4})|$).

Essayer toutes les possibilités incluant les 10 paiements choisis.

Si au moins l'une des possibilités permet le règlement.

Alors choisir l'ordre de paiement qui maximise la valeur de paiement.

Si non

On désactive tous les paiements de la banque défaillante (on suppose que ce soit la banque A).

On prend les paiements de la banque A (a_i) $i=1\dots N$ de plus grand a_1 au plus petit a_N .

Un paiement a_k peut être :

Si $a_k >$ fond virtuel de la banque A.

Alors ce paiement ne peut pas être réactivé.

Si non

On active a_k .

On passe au prochain paiement.

Fin.

Régler tous les paiements activés.

3. Optimisation multilatérale

Dans cette partie on se concentre sur l'optimisation multilatérale. Cette fois-ci, tous les participants et tous les paiements enfilés sont considérés simultanément.

3.1. Modèle de Renault et Pecceu 2007 [HAR, 07]

Consiste à trouver le plus tôt possible un ensemble de paiements avec une très grande valeur, et qui peuvent être réglés simultanément.

Le principe est de mettre le flux existant dans le centre du réseau formé par les participants. Pour cela sont désactivés les paiements enfilés concernant les **participants périphériques**.

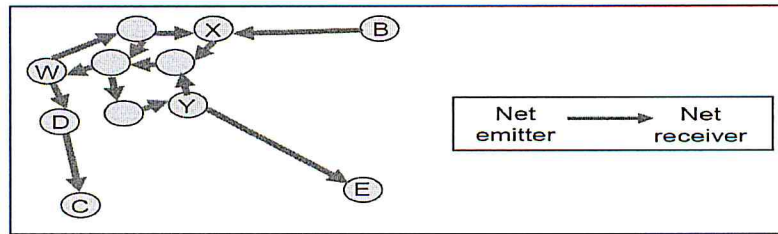


Fig. IV.1 Réseau formé par les participants [HAR, 07].

Il existe 2 types de participants périphériques :

- Les émetteurs net comme B (B est l'émetteur net, car la valeur cumulé des paiements de B vers X est plus grande que celle de X vers B, à noter que B est la source de réseau) .la raison pour laquelle les paiement entre X et B sont tenu dans la file est, que B n'a pas assez de liquidité pour régler sa position net ;et comme B ne peut pas avoir de liquidité ,donc les paiements entre B et X ne peuvent jamais être réglés.
- Les receveurs net comme(C et E); les liquidités qui reçoivent ne peuvent pas être utilisées pour un autre règlement dans le réseau.

Il a été conclu que quelque soit la position (émetteur net ou receveur net), les participants périphériques ont un impact négatif sur le réseau. On peut donc les supprimer avant d'exécuter l'algorithme; pour mieux l'optimiser.

3.2.Algorithme Multilatéral Greedy Las Vegas

Activer tous les paiements.

1. essayer de régler tous les paiements simultanément (le principe de tout ou rien).
2. éliminer tous les paiements qui sortent ou entrent aux participants périphériques.
3. Quand on a un participant avec un fond virtuel négatif.
 - 3.1. On arrange toutes les banques qui possèdent un fond négatif (c'est-à-dire les banques débitrices) dans un ordre croissant selon le nombre de participants qui sont sollicités par chacun d'eux. Ensuite, on les ordonne par leur valeur de déficit selon l'ordre croissant (après on débute avec les participants qui émettent des contreparties vers un seul participant).
 - 3.2. La banque prise i à un fond virtuel négatif B_i .

Désactiver tous ses paiements sortants, puis les ordonner selon l'ordre décroissant de leur valeur. Chaque paiement peut être réactivé, mais il doit garantir que le fond virtuel reste positif.

On a donc pour le paiement numéro 1 de la banque i , P_i^l .

a. Si $P_i^l > B_i$ alors le paiement numéro 1 ne peut pas être réactivé.

b. Si non

si $R_i^1 < B_i$ alors le paiement numéro 1 est réactivé. (ou $R_i^1 = \sum_{k \geq l+1} p_i^k$ est la valeur cumule des paiements plus petit que P_i^l).

c. Si non le paiement P_i^l est réactivé avec une probabilité égal au $\min(\frac{P_i^l}{R_i^1} b^+ ; 1)$, ou : $b^+ = \frac{m^+ + m^-}{2m^+ + m^-}$ et

$$b^- = \frac{m^+ + m^-}{m^-} b^+.$$

m^+ (resp. m^-) est le nombre des participants qui reçoivent les paiements depuis la banque i et qui ont un fond virtuel positif respectivement négatif.

$b^+ = b^-$ c'est le receveur du paiement a un fond virtuel négatif, si non : $b^+ = b^-$.

Fin

4. quand tous les paiements ont un fond virtuel positif, tous les paiements activés sont réglés.
Réactiver tous les paiements désactive (incluant les participants périphériques).

On se concentre sur toutes les relations bilatérales entre les participants, depuis le plus équilibré au plus déséquilibré ensuite on exécute l'algorithme Las Vegas Greedy [HAR, 07].

4. Modélisation de la dynamique des systèmes de paiement :

Beyeler et al. Ont fait en 2007 une étude qui consiste à trouver une solution à la surcharge des paiements enfilés, qui attendent leur rôle de règlement. Pour remédier ce problème il faut trouver une autre source de liquidité.

Le modèle de Beyeler et al ne comprend que les processus essentiels d'un système de paiement et sa liquidité sur le marché. Un ensemble de banques échange de paiements à travers un système de paiement unique et commun. Tous les paiements sont effectués seulement sur les liaisons d'un réseau. Les clients des banques demandent au hasard de faire un paiement à une unité aux banques voisines connectées.

Les banques se coopèrent réflexivement: ils soumettent le paiement si le solde de leur compte au sein du système de paiement permet l'opération, sinon de mettre l'instruction dans sur une file d'attente pour la régler plus tard. Si une banque a reçu des instructions de paiement dans sa file d'attente, et elle reçoit un versement juste après :

Cela permet de supprimer une instruction de la file d'attente et soumettre un paiement à son tour. Si la banque qui a reçu un tel paiement a également d'autres instructions dans la file d'attente, elle peut effectuer un autre paiement, et ainsi de suite. De cette façon un seul versement initial effectué par une banque peut causer de nombreux paiements à effectuer des files d'attente qui sont en manque de liquidité. Il s'agit d'un exemple de cascade de processus. Les statistiques sur ces cascades de règlement sont un indicateur de l'interdépendance des banques, et dans ce modèle elles sont contrôlées par deux paramètres: la liquidité globale et la conductance du marché [HAR, 07].

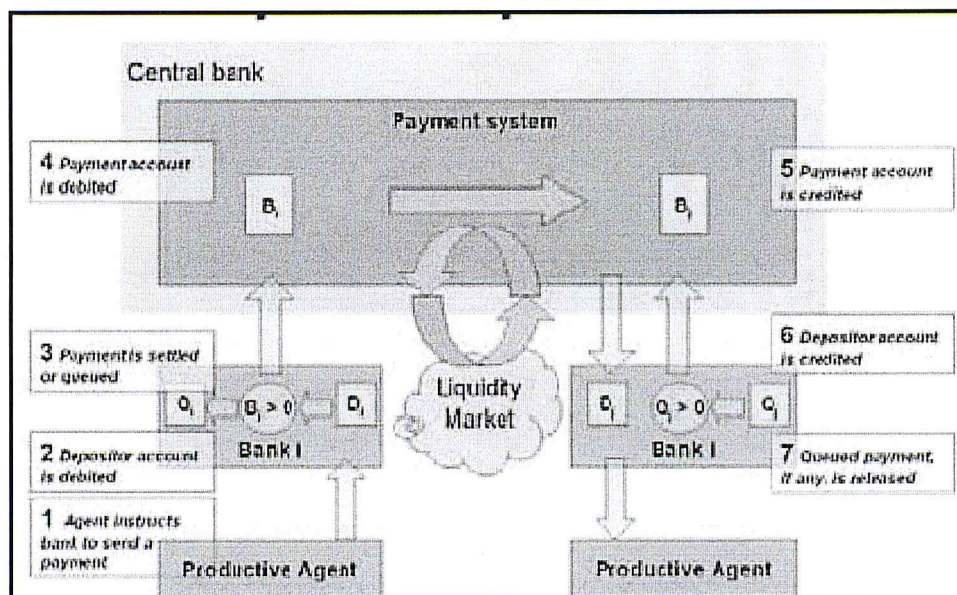


Fig. IV.2. Un modèle simple d'un système de paiement [HAR, 07].

En l'absence d'un marché de liquidités, seulement une liquidité importante (grande valeur) permet aux banques de fonctionner d'une façon indépendante. La réduction de la liquidité augmente la probabilité qu'une banque donnée aura épuisé son équilibre et commence à enfile les paiements. Une banque qui a épuisé son solde doit attendre un paiement entrant de l'un de ses voisins. Lorsque la liquidité est faible, la capacité d'une banque à traiter ses paiements devient attacher à la capacité des traitements de ses voisins.

Figure IV.3.a : montre comment la corrélation entre l'arrivée des instructions et la soumission des paiements se dégrade dans le modèle quand la liquidité est réduite (1: haute de liquidité; 2: liquidité à moyen; 3: faible liquidité). Une cascade de règlement, qui est la libération des paiements en file d'attente à la suite d'un paiement peut comprendre des centaines de paiements en file d'attente, comme il est illustré dans la Figure IV.3.b : Pour découvrir comment les marchés de liquidités réduisent le couplage (dépendance) entre les voisins et ainsi réduire la congestion ; les transactions de marché ont été représentée comme un processus de diffusion où l'équilibre d'une banque joue le rôle d'une énergie potentielle ou de pression. Les banques ayant des soldes élevés ont tendance de contribuer à la liquidité du marché, tandis que les banques avec de faibles soldes ont tendance à attirer des liquidités du marché. Il n'y a pas de décision ou de fixation des prix dans ce modèle simple du marché, mais il reflète deux éléments essentiels d'un

véritable marché: la liquidité des banques avec un fond surplus découle les banques qui ont besoin de fonds, et à noter aussi que la liquidité peut découler de n'importe quelle banque à une banque - les flux ne se limitent pas aux liens du réseau de paiement. Il crée une voie séparée pour le flux de liquidités. La facilité d'écoulement de liquidité à travers le marché qui est décrit par un seul paramètre de conductance.

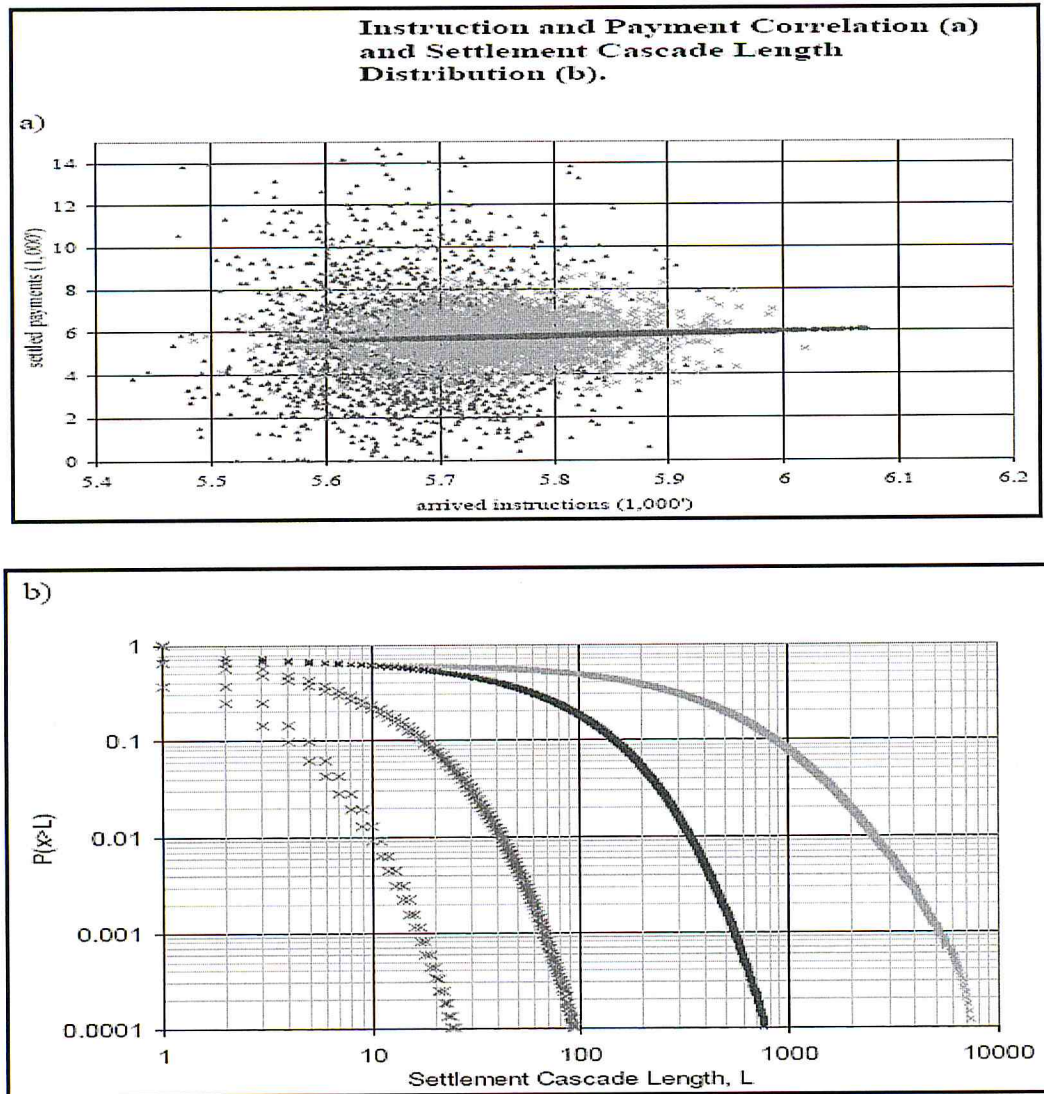


Fig. IV. 3. Instruction et corrélation de paiement (a) et distribution Longueur de règlement de Cascade (b)[HAR, 07].

Avec un marché de liquidités inclus dans les échanges interbancaires, la dépendance entre les banques pour régler un certain nombre d'instructions de paiements est affaiblie et la taille des cascades de règlement est réduite. Le taux

des flux de liquidités par le marché par rapport à la vitesse d'écoulement par le biais du système de paiement a été étonnamment faible.

5. Modélisation du comportement d'un système bancaire par multi agent :

Galbiati et Soramäki [HAR, 07] essaient de comprendre comment interagissent et réagissent les participants face à l'adversité opérationnel. ceci va aider les operateurs et les régulateurs à concevoir des contre mesure, l'élaboration d'une politique et fournir une consistance d'urgence si nécessaire.

Galbiati et Soramäki ont développé un modèle dynamique multi-agent d'un système de paiement interbancaire où les paiements sont réglés sur la base de fonds de pré-engagés. Dans ce modèle les banques choisissent leur niveau de fonds engagés sur la base de la maximisation du profit privé. Le modèle consiste en une séquence de jours de règlement. Chacun de ces jours est un jeu de déplacement simultané, dans lequel chaque banque choisit le montant des liquidités à s'engager pour le traitement des paiements et reçoit un gain stochastique (aléatoire). Les gains sont déterminés au moyen de simuler les jours de règlement avec les montants de liquidités choisis par les banques. Pour que les instructions soient réglées par les banques, ils arrivent sur la base d'un processus de Poisson. Par le procédé de chaque maximisation de profit, les banques ajustent leurs demande de liquidités vers le haut (réduction des délais) quand le cout de délai augmente, et vers le bas (l'augmentation des retards).

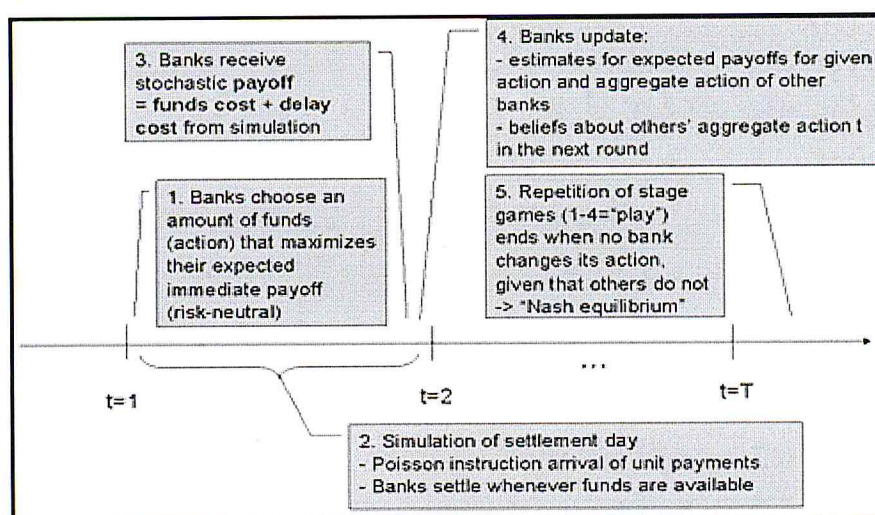


Fig. IV.4. Aperçu d'un modèle multi-agents d'apprentissage d'un système de paiement

[HAR, 07]

6. Conclusion

Les travaux de recherches présentés plus haut ont beaucoup apporté pour le comportement envers les opérations à traiter par les systèmes de compensation interbancaires.

On s'est inspiré de ces modèles et nous avons essayé d'appliquer deux idées primordiales dans notre système concernant l'optimisation multilatérale.

- Favoriser les opérations qui apportent le plus de gains pour la banque, c'est à dire dans une situation où une banque devient débitrice, nous allons favoriser les opérations qui offrent un gain (les opérations à crédit comme par exemple : les chèques si la banque est remettante ou bien les virements si la banque est destinataire) pour diminuer la valeur de débit.
- Renforcer les opérations de paiement qui apportent un gain considérable. Dans ce cas, on renforce les règles (de traitement des opérations de paiement) du système de classeurs en donnant une priorité aux paiements qui apportent le plus de gains pour la banque.

Les fonctionnalités d'optimisation que nous allons apporter n'existent pas dans le système de paiement Algérien actuel.

Mais avant d'aborder la conception de notre système, nous présentons le système de télé-compensation interbancaire algérien.

1. L'entreprise accueillante 'CENTRE DE PRÉ-COMPENSATION INTERBANCAIRE'

La société a été créée par l'assemblée générale constitutive en date du 04 août 2004 sous forme de filiale dont le capital de cinquante et un millions de dinars (51.000.000 D.A) appartient intégralement à la Banque d'Algérie. Ses statuts ont été notariés auprès du Maitre REZZAG BARA Ali.

Une assemblée générale extraordinaire a eu lieu le 05 août 2004, autorisant l'augmentation du capital de la société à cent millions de dinars (100.000.000 D.A) et son ouverture à l'ensemble des banques et Algérie Poste à parts égales, soit deux millions de dinars (2.000.000 D.A) par actionnaire.

Onze (11) banques ont répondu à cette demande d'augmentation.

Une assemblée générale a été tenue le 31 octobre 2004 afin de :

1. Constaté l'augmentation du capital de vingt deux millions (22.000.000) de Dinars pour le ramener à 73 millions de Dinars.
2. Élargir la représentation au niveau du Conseil d'Administration à quatre (4) banques actionnaires parmi les onze.

La composition du Conseil d'Administration se présente comme suit :

1. Cinq (5) sièges pour la Banque d'Algérie.
2. Quatre (4) sièges pour les banques actionnaires.

L'assemblée générale extraordinaire de la société a décidé, en date du 09 juin 2005, d'augmenter le capital social de huit cent millions de dinars (800.000.000 D.A). Ont souscrit à cette augmentation dix-sept (17) banques et Algérie Poste pour un montant global de 787.950.000,00 D.A. Ainsi, le capital du C.P.I. est passé de 75.000.000,00 D.A à 862.950.000,00 D.A sans aucun changement dans la composition du Conseil d'Administration.

Le Centre de Pré compensation Interbancaire est une société par actions de services interbancaires. Il a pour objet la mise en œuvre et l'exploitation d'un système automatisé de compensation des paiements de masse dématérialisée, le système utilisé actuellement est le **Système d'échange et de Compensation des Paiements de Masse** (appelé par la suite S.C.P.M).

2. Système d'échange et de compensation des paiements de masse

L'architecture et la conception du SCPM doivent assurer au système sa robustesse et haute disponibilité, sécurité, exploitabilité et interfaçage avec les autres systèmes de la Banque Centrale.

2. 1. Vue technique du SCPM

Le Système de Compensation de Paiement de Masse du CPI consiste en la mise en oeuvre de :

- ✓ Un site de compensation principal.
- ✓ Un site de compensation de secours à chaud : qui est synchronisé en temps réel avec le site principal.
- ✓ Un site de compensation de secours à froid : qui est maintenu à jour par rapport au site principal à l'issue de chaque journée de compensation.¹⁴.
- ✓ Une solution de raccordement des participants : qui relie le S.C.P.M aux UAPs, qui sont eux même interfacés avec les Systèmes d'Informations des Banques.
- ✓ Une solution d'interfaçage avec les systèmes externes de la banque centrale : où tout système externe (RTGS et CIP) est relié au Serveur d'Échange.

Les systèmes centraux (SCPM) sont des ensembles applicatifs qui résident chacun dans un cluster UNIX comportant une base de données Oracle 10g.

Les sites de secours, identiques au site principal, permettent la continuité de service en cas de défaillance du site principal.

Un participant au S.C.P.M est un établissement bancaire qui possède un compte de règlement au niveau de la Banque d'Algérie qui présente ses opérations de paiement en son nom propre et pour laquelle le Système A.T.C.I. a calculé un solde multilatéral.

Deux types de participants sont définis :

- **Les Participants Directs :** qui possèdent une U.A.P. reliée au Système A.T.C.I.,
- **Les Participants Indirects :** qui utilisent l'U.A.P. d'un Participant Direct.

La notion de « Participant » n'est qu'une notion technique qui définit une entité logique de raccordement.

Un Participant direct est

- **Présentateur de flux** : il possède une U.A.P. qui présente des flux Aller Compensation et des fichiers d'images, pour son propre compte et pour le compte des participants indirects qu'il représente (dans des fichiers distincts).
- **Destinataire de flux** : il reçoit via son U.A.P. des flux Retour Compensation et des fichiers d'images, pour son propre compte et pour le compte des participants indirects qu'il représente (dans des fichiers distincts).

Note : Le Participant Direct n'est pas responsable financièrement des Participants Indirects qu'il représente ; il agit en tant qu'opérateur technique pour ses participants indirects.

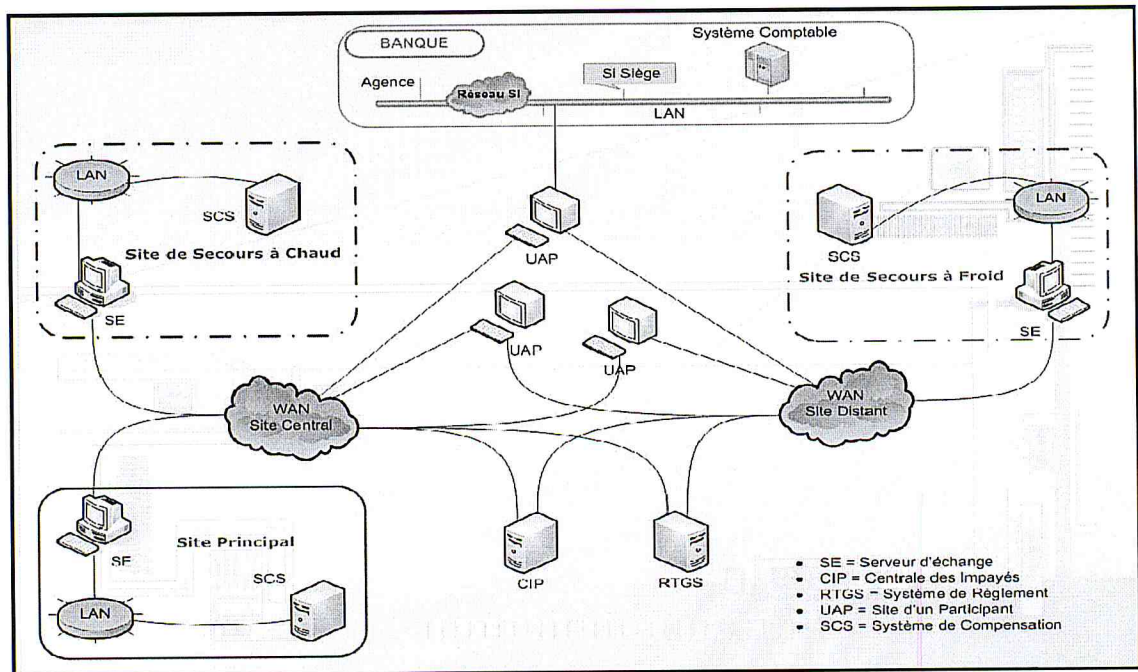


Fig. V.1: Diagramme basique du réseau du S.C.P.M.

2. 2. Vue fonctionnelle du SCPM

Une plateforme de raccordement UAP est mise à la disposition de chaque Participant Direct pour qu'il puisse communiquer avec le S.C.P.M.

Le S.C.P.M se compose d'un serveur de traitement et d'un serveur d'échange. Ce dernier est le système qui permet l'acquisition et la révision des fichiers échangés entre les Banques et le S.C.P.M. Ces fichiers contiennent des opérations de paiement de tout type (chèques, prélèvements...), ou des informations concernant l'état du système et de ses composants. D'autre part, le serveur d'échange permet également l'alimentation au fil de l'eau du serveur de secours à chaud (synchronisation totale des deux sites).

La sécurité et la fiabilité des échanges sont garanties au moyen de protocoles qui utilisent des standards internationaux tels que les Certificats X509 et le protocole SSL V316, cela permet de maintenir le système dans un haut niveau de disponibilité, d'une façon où la resynchronisation des échanges soit totale et sans aucune perte d'information.

Les banques peuvent également utiliser des supports physiques tels qu'un CD ou DVD pour leurs échanges, et cela éviterait le blocage des processus de Télé-compensation en cas de rupture de leur liaison télécom.

Les principaux services fournis par le S.C.P.M sont les suivants

- ✓ Support de tout moyen de paiement : virements, prélèvements, chèques...
- ✓ Gestion des fichiers soumis par les participants.
- ✓ Gestion des rejets d'opérations de paiement.
- ✓ Gestion des fichiers générés et diffusés par le S.C.P.M à l'issue de chaque séance de compensation par participant destinataire.
- ✓ Support des Annulations partielles ou totales des ensembles d'opérations de paiement.
- ✓ La distribution de messages d'information vers l'ensemble des participants, l'échange de messages entre un participant et le S.C.P.M et vice versa, le routage automatique vers la Centrale des Impayés des déclarations d'incidents de paiement.
- ✓ Gestion du risque financier.
- ✓ Calcul des soldes de règlement.
- ✓ Fourniture d'informations techniques et financières aux participants.
- ✓ La configuration d'une journée de configuration à plusieurs séances.

2.3. Fichiers d'Entrées / Sorties

Au niveau du S.C.P.M, on voit beaucoup de fichiers plats d'E/S entre les UAP des participants et le système. Seuls quatre (4) fichiers ont un lien avec les opérations monétaires en état d'entrée ou de sortie.

Le S.C.P.M reçoit des Remises Aller générées et émises par les UAP. Il permet d'utiliser trois (3) types différents de Remise Aller : ICOM1, ICOM2 et ICOM3.

Un fichier de type ICOM1 : qui est un fichier constitué d'opérations de paiement généré par une U.A.P. pour le compte d'un participant (direct ou indirect) et transmis au Système A.T.C.I. ; les Remises Aller d'opérations peuvent contenir différents types d'opérations de paiement.

Un fichier de type ICOM2 : qui est un fichier constitué de rejets d'opérations de paiement généré par une U.A.P. pour le compte d'un participant (direct ou indirect) et transmis au Système A.T.C.I. ; les Remises Aller de rejets d'opérations peuvent contenir différents types de rejets d'opérations de paiement.

Un fichier de type ICOM3 : est une Remise Aller qui est une annulation totale ou partielle de Remise Aller (sous-remises, opérations).

Les Remises Aller sont constituées en sous-remises (chacune adressée à une seule banque de contrepartie pour un type d'opération unique).

Une Remise Retour d'Opérations : est un fichier créé par le S.C.P.M pour un Participant Direct ou Indirect suite à la fin d'une séance de compensation. Le type du fichier de remise de retour est : **OUTGO**.

Un OUTGO est un fichier constitué de sous-remises d'opérations de paiement ou de rejets d'opérations qui ont été reçues et acceptées dans les Remises Aller d'Opérations émises par les autres participants durant la même séance de compensation. Chaque sous-remise est constituée d'opérations de même type (même code opération) provenant d'une seule banque remettante.

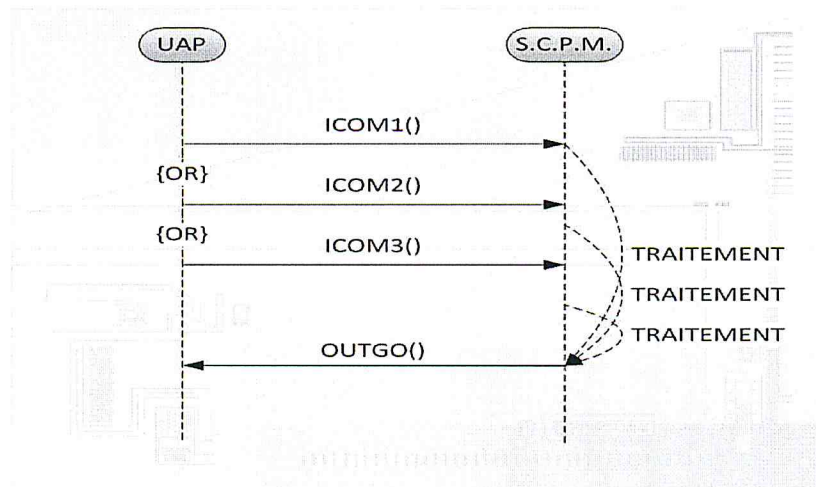


Fig. V.2 Exemple des fichiers plats porteurs d'opérations en entrées / sorties.

3. Présentation des flux des échanges entre les U.A.P. et le système A.T.C.I

Les U.A.P. échangent avec le Système A.T.C.I. des flux de différentes natures :

- Flux Monétaires.
- Flux Financiers.
- Fichiers Images.
- Messages d'Information.
- Flux de Synthèses.
- Flux de Statistiques.
- Flux Techniques.

Dans notre étude on s'intéresse à 2 flux d'échanges :

3.1. Flux de statistiques

Les différents types de flux de statistiques sont :

- **Statistiques Aller (STATI)** : les statistiques aller correspondent aux mêmes informations que les synthèses aller mais pour une période de plusieurs journées de compensation.
- **Statistiques Retour (STATO)** : les statistiques retour correspondent aux mêmes informations que les synthèses retour mais pour une période de plusieurs journées de compensation.

3.2. Flux d'alarme

Les flux d'alarme générés par le Système A.T.C.I. durant la période d'échange permettent d'avertir tous les participants d'un événement. En particulier, ces flux sont utilisés afin de prévenir les participants d'un décalage dans la période d'échange ou de la suspension ou exclusion d'un participant ou de l'annulation d'une journée de compensation qui est déjà commencée. Ce flux est également généré par le Système A.T.C.I. lors du traitement d'une Remise Aller (envoyée par un participant remettant) qui entraîne le dépassement du seuil d'alerte de limite financière d'un (ou plusieurs) participant(s) destinataire(s). Le flux contient la position du participant destinataire mise à jour. A l'initialisation de la journée de compensation, des flux d'alarme peuvent également être générés par le Système A.T.C.I. pour tous les participants qui ont une position initiale qui dépasse leur seuil d'alerte de limite financière.

4. Organisation de la journée de compensation

La journée de compensation se compose de trois périodes :

- La période d'échange qui comporte trois séances de compensation.
- La période de règlement.
- La période d'exploitation.

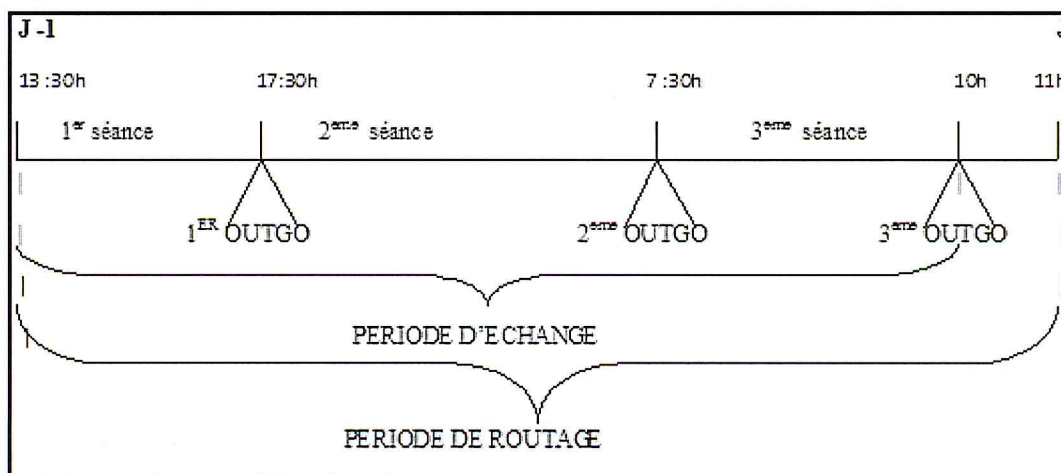


Fig. V.3 Les séances de la journée de compensation.

5. Classification des moyens de paiement

Nous distinguons deux types de moyen de paiement.

5.1. Les moyens de paiement à crédit ; Ce sont les moyens qui aident la banque pour garder sa stabilité ils ont un impacte positif sur le participant. Il s'agit des chèques, des prélèvements, des effets et des cartes.

5.2. Les moyens de paiement à débit ; Ce sont les moyens qui ont un impacte négatif sur le participant. Il s'agit des virements.

6. classification des banques

Nous distinguons deux types de classification.

6.1. Classification selon la nature des banques

Nous avons les banques publiques et les banques privés.

6.2. Classification selon le budget de la banque

Dans notre étude nous intéressons plus à ce type de classification d'où nous avons trois types de participant :

Grande	Moyenne	Petite
C.P.A	Citibank	HSBC Algeria
B.E.A	Natixis Algérie	Arab Bank plc
Trésor public	C.N.E.P Banque	Fransabank El Djazair
B.A.D.R	Banque El Baraka d'Algérie	Housing Bank
B.N.A	Algeria Gulf Bank	Banque d'Algerie
B.D.L	A.B.C	Al Salam Bank Algeria spa
Société Générale Algérie	Trust Bank	Calyon Algérie spa
B.N.P.Paribas		
Algérie Poste		

Tab. V.1 : Classification des banques.

7. Réforme du système de paiement les systèmes ATCI (Algérie Télé-Compensations Interbancaire)

Les objectifs

✚ Pour les pouvoirs publics

- ◆ Réduire les délais de traitement des opérations.
- ◆ Accélérer et sécuriser les échanges.
- ◆ Réduire les coûts et délais de traitement.
- ◆ Améliorer la fluidité de la circulation des moyens de paiement.
- ◆ Relever la rentabilité des banques.
- ◆ Améliorer la qualité de leurs services.
- ◆ Développer la bancarisation.
- ◆ Renforcer le rôle du secteur financier dans l'économie et sa stabilité.
- ◆ Assurer la traçabilité des opérations.

✚ Pour les autorités de régulation

- ◆ Réduire les délais de recouvrement des paiements scripturaux.
- ◆ Mettre en œuvre des systèmes de paiements sécurisés.

✚ Pour le secteur bancaire

- ◆ Améliorer l'image des services bancaires.
- ◆ Éliminer les risques opérationnels.
- ◆ Réduire les coûts de traitements.

8. Description des besoins

Après une petite analyse descriptive du CPI et de son système nous pouvons citer les remarques suivantes :

- La description du système (BDD et partage de fichier) doit être grossière en raison de la confidentialité.
- Notre application ne doit ni copier ni modifier les schémas des bases de données du SCPM en raison de droit d'auteur.
- Jusqu'au aujourd'hui le SCPM ne se concentre pas sur l'opération elle-même mais sur le traitement globale des échanges interbancaire de ces

opérations. dans notre travaille nous allons prendre en considération l'opération elle-même.

- Nous avons remarqué aussi que le système CPM envoi que des alarmes aux participants qui ont dépassé leur limite financières sans aucune intervention sur les échanges effectuer au-delà de la limite financière.

9. Conclusion

Le système de compensation de masse est très bien structuré techniquement et fonctionnellement. Il impose aux participants d'uniformiser leurs façon de représentation des opérations de paiement à travers un format de fichier bien précis appelé (remise aller). Les délais de dépôt sont bien déterminés ; ainsi que la période de règlement des paiements. De plus le système fournit plusieurs services qui facilitent l'adaptation des banques à ce nouvel dispositif. Le système contrôle les banques et les avertis par des flux d'alarme ; mais cela reste insuffisant ; car il lui manque toujours cette couche d'intervention, afin de décider ce qui doit passer ou pas.

1. Introduction

Notre projet consiste en la conception et la réalisation d'un système de suivi intelligent des échanges dans un système de compensation interbancaire en utilisant la technologie multi-agent et les systèmes de classeurs pour le modèle de raisonnement des agents. L'objet de notre étude porte sur le traitement des soldes présentés par les différentes banques et leur répercussion sur la stabilité du système ATCI. L'objectif principal est d'introduire des stratégies qui aident les banques à choisir leur quota d'opérations selon leur statut (remettante ou bien destinataire) et par rapport au type d'instrument (à crédit ou à débit), et d'assurer une visibilité permanente sur la situation financière des banques, pour garantir la bonne circulation de liquidité journalière dans le système ATCI.

Pour remplir notre objectif, nous construisons un modèle multi-agent qui joue le rôle d'une couche supplémentaire qui travaille en collaboration avec le système central ATCI, pour aider les banques à prendre des décisions fiables et d'empêcher toutes les entités (banques) à tomber dans des situations de défaillance. Notre système prend en charge tous les échanges interbancaires présentés par les banques au système ATCI ; et essaie de contrôler les opérations de plusieurs journées de compensation où un ensemble de banques effectuent des opérations de paiement de différentes valeurs et natures, pour aider les banques à prendre des décisions. Du point de vue multi-agent, l'aspect apprentissage est décrit par la succession des journées, où l'agent va apprendre et par conséquent prendre des décisions qui sont meilleures.

Nous allons donc utiliser un processus d'apprentissage par renforcement permettant aux banques d'adapter leurs actions au cours du temps.

Nous nous concentrons sur le suivi du solde de chaque banque. Comme résultats, notre système optimise le comportement de chaque banque envers ses opérations de paiement. Ensuite, on obtient plusieurs paramètres qui décrivent la situation de chaque banque dans le système ATCI.

Nous commençons ce chapitre par la présentation des deux méthodes de conception (puisque nous travaillons sur un mélange de système d'information et un système multi agent) l'une comme étant une approche pour les systèmes d'information c'est la méthode OMT et l'autre comme approche pour les systèmes multi agent c'est la méthodologie *Voyelles*.

2. La méthode OMT

2.1.Introduction

La méthode OMT (Object Modeling Technic), que l'on peut traduire en français comme Technique de Modélisation Objet, est sans aucun doute la plus répandue des méthodes orientées objet. Mise au point dans le Centre de Recherche et Développement de la société Général Electric à New-York (USA) à la fin des années 80 elle est vulgarisée en 1991 dans un ouvrage appelé « Object-Oriented Modeling and Design » écrit conjointement par Rumbaugh J, Blaha M, Premerlani W et Eddy F, elle est suivie ensuite par OMT-2 en 1993 Cette méthode est basée sur une modélisation entités/rerelations étendue afin d'intégrer les concepts objets.

2.2.Présentation générale de la méthode OMT

2.2.1. Esprit de la méthode

- Modélisation d'objets du monde réel.
- Utilisation de ces objets pour concevoir un système indépendamment des langages d'implantation.

2.2.2. Avantage

- Meilleure compréhension des besoins.
- Spécification et conception plus précise.
- Meilleure maintenabilité des systèmes réalisés.

2.3.La démarche générale de développement :

La méthode OMT se décompose en trois grandes étapes, l'analyse, la conception et l'implémentation.

2.3.1. L'analyse

L'analyse qui consiste à élaborer l'ensemble des modèles suivant les trois axes statique, dynamique et fonctionnel, est effectuée en collaboration avec le client car les spécifications initiales sont rarement complètes ou correctes. Un bon modèle doit pouvoir être compris par l'utilisateur qui bien qu'expert de son domaine n'est généralement pas un spécialiste des outils informatiques.

2.3.1.1. Modèle statique

Le modèle statique : représente la structure statique par l'ensemble des classes et des relations (association, agrégation, généralisation, spécialisation) qui les relient, ainsi que les attributs, contraintes, et opérations.

Le diagramme de classes exprime de manière générale la structure statique d'un système en représentant les classes et les relations entre ces classes.

2.3.1.2. Modèle dynamique

Construction du modèle dynamique qui montre les différentes séquences d'exécution des opérations par l'élaboration des diagrammes d'activité.

Le modèle dynamique n'a d'intérêt que dans une application interactive ou temps réel bien que ce dernier comporte des spécificités propres aux contraintes qui en découlent.

2.3.1.3. Modèle fonctionnel

Construire le modèle fonctionnel pour permettre une meilleure visibilité de ce qui doit être fait par le système. Le modèle fonctionnel décrit en particulier les transformations des données par rapport aux fonctions du système, la nature et le circuit des échanges au moyen de représentation de flux de données, en dehors de toute considération temporelle.

Cas d'utilisation (Use Case)

Les fonctionnalités du système sont décrites dans le modèle des besoins (use case view) par un ensemble de cas d'utilisation. C'est une description générique d'une utilisation du système (les scénarios en merise). Elaboré en langage naturel avec le client, puis représenté graphiquement de la façon la plus explicite possible, il offre l'avantage de préciser le contenu souvent flou du cahier des charges et évite aux informaticiens de s'éloigner des besoins réels de l'utilisateur.

Comme pour UML, un cas d'utilisation permet de décrire l'interaction entre le système et un acteur externe. Le modèle de cas d'utilisation comprend les acteurs, le système et les cas d'utilisation eux-mêmes.

Le succès des Use case réside dans leur simplicité d'expression, particulièrement accessible aux utilisateurs désireux de formaliser leur activité, mais ils ne répondent pas à toutes les attentes. Notamment ils ne font pas

apparaître toutes les ressources informationnelles exploitées par les fonctions, elles ne sont pas hiérarchisées. Par ailleurs ils ne prennent pas en compte la complexité du système, ne montrent pas les interactions globales utilisateurs/système et ne mettent pas en évidence le lien avec les autres diagrammes.

2.3.2. Conception du système

Le modèle de conception s'occupe de détail de plus bas niveau que ceux du modèle d'analyse, ce dernier se rapproche plus des contraintes liées à l'implémentation.

Les modèles produits dans la phase d'analyse sont ainsi améliorés afin de créer le diagramme d'architecture général du système. Un découpage en sous-système puis en modules peut-être réalisé.

2.3.3. Implémentation

Etape d'implémentation concrète qui sert à produire le logiciel défini dans la phase de conception. C'est la phase de programmation.

3. Méthodologie de conception du SMA

Parmi les méthodes qui couvrent le mieux le cycle de développement d'un système multi-agent, que nous avons citées dans le chapitre des systèmes multi-agents, nous avons choisi la méthodologie *Voyelles* pour modéliser notre système. Ce choix se justifie comme suit :

- Cette méthode repose sur la décomposition du système en cinq dimensions : **Agent**, **Environnement**, **Interaction**, **Organisation**. Cette décomposition permet de modulariser le système, simplifier sa construction et offrir une meilleure réutilisation du code.
- *Voyelles* n'est couplée à aucune notation ni plateforme, ce qui offre la possibilité d'utiliser le langage AUML (Agent Unified Modeling Language), qui est une extension du langage UML, dans la phase de conception du système et la plateforme Jade pour implémenter le système.
- *Voyelles* repose sur des principes purement multi-agents.

Le processus de développement avec la méthode *Voyelles* (AEIO) comporte trois étapes essentielles :

- **Analyse** : consiste à identifier les cinq composantes : Agent (A), Environnement (E), Interactions (I), Organisation (O), Utilisateurs (U).
- **Conception** : il s'agit de choisir les modèles opérationnels des composantes.
- **Implémentation** : consiste en l'instanciation des modèles en utilisant des plateformes et des langages choisis.

4. Analyse du système

Dans cette partie nous allons décrire les caractéristiques du système à réaliser, les composants macros du système.

4.1. Spécification du système :

Selon l'objet de notre étude, notre modèle cherche à apporter une compréhension sur les choix des opérations des banques qu'elles affectent quotidiennement au règlement de leurs paiements dans le système ATCI. Cela nous oblige à restreindre le champ d'étude aux caractéristiques du système ATCI uniquement essentielles à l'objet de cette étude.

La modélisation du système ATCI s'articulera donc autour de deux composantes principales:

- Le système central qui constitue le noyau du système ATCI et qui reçoit et traite les opérations interbancaires.
- Les banques qui échangent des opérations entre elles à travers le système central.

4.2. Analyse des composants du système

4.2.1. Identification des agents du système

Pour identifier les agents de notre système. Nous avons deux types d'agents : un « **agent ATCI** » et plusieurs « **agents Banques** ».

4.2.1.1. Agent ATCI

Notre application nécessite d'avoir un agent de règlement au niveau du système central. Dans notre modèle, l'agent ATCI représente le système central de l'ATCI, ce dernier est au centre des paiements interbancaires, il est chargé de gérer les opérations des banques et d'effectuer les transferts interbancaires.

Selon les spécifications de notre modèle, cet agent devra remplir les rôles suivants :

- Déclenchement d'Ouverture d'une Journée de Compensation.
- Alimenter la BDD locale.
- Génération des fichiers XML.
- Mise à jour de la base de données locale.
- Déclenchement de la Fermeture de la Journée de Compensation.

4.2.1.2. Agent Banque (AB)

Les agents banques constituent les acteurs principaux du système car ils sont les seuls à être concernés par les opérations. Tout au long de la journée, ils s'échangent des opérations. Ainsi, les activités de chaque agent banque sont les suivantes :

- Préparation des Paramètres du CS (EP, TB,..).
- Déclencher le Cycle CS et prendre la décision.
- Lecture des opérations à partir de la BDD locale.
- Calculer le Solde à Chaque Opération.

4.2.2. Schéma général du système

Le modèle comprend un système multi-agent qui constitue l'élément principal permettant le suivi et la prise de décisions par rapport aux échanges interbancaires. L'agent ATCI a accès à la BDD Système contenant les données sur les opérations échangées entre les banques, et alimente la base de données locale. Les agents banques sont en interaction permanente entre eux à travers l'envoi des opérations de paiement.

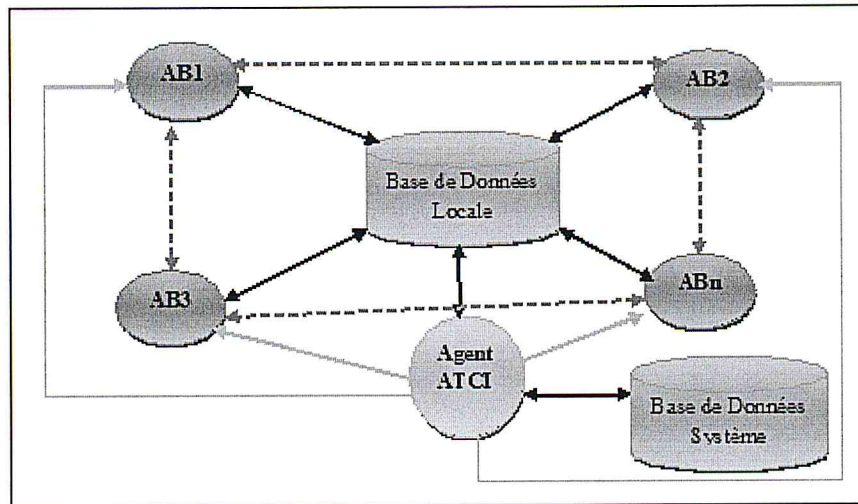


Fig.VI.1 Schéma général du système.

Comme le schéma ci-dessus le démontre notre système est composé d'un agent ATCI et des agents banques et deux bases de données, une est appelée système, elle est exploitée par l'agent ATCI pour l'extraction de données et l'autre appelée Locale qui l'alimente. L'agent ATCI est en interactions permanente avec la base de données système et la base de données locale ; il transmet aussi des informations pour les autres agents. Les agents banques entourent l'agent ATCI et échangent des messages entre eux.

4.2.3. Environnement

L'environnement du système multi-agent contient l'ensemble des objets manipulés par les agents du système. Dans notre cas, l'environnement se constitue d'une base de données contenant les informations sur les banques, la limite financière de chaque banque et les opérations échangées. Nous verrons plus loin la description des classes de la BDD locale dans la phase de conception.

4.2.4. Interactions

Nous distinguons deux types d'interaction :

- ✦ Les interactions que l'on observe dans notre modèle sont celles qui se passent entre les agents banques. Ces interactions sont :
 - l'agent représentant la banque émettrice doit informer l'agent qui représente la banque réceptrice après l'acceptation d'une opération de nature (chèque, prélèvement, carte, effet) pour la mise à jours du solde

(ajout du montant de l'opération), qui est nécessaire au traitement des autres opérations.

- l'agent représentant la banque remettante doit informer l'agent qui représente la banque réceptrice après l'acceptation d'une opération de nature (virement) pour la mise à jour du solde (ajout du montant de l'opération), qui est nécessaire au traitement des autres opérations.

⚡ Les interactions entre agents banques et agent ATCI :

L'agent ATCI doit informer les agents banques de l'ouverture et de la fermeture d'une journée de compensation.

Les diagrammes d'interactions nous les verrons un peu plus loin.

4.2.5. Organisation

L'organisation est la structure décrivant comment les membres du système interagissent entre eux. Dans notre système, on voit que les agents banques et l'agent ATCI sont regroupés autour de la BDD locale. Les communications se font entre les agents banques d'une part et d'autre part entre l'agent ATCI et les agents banques.

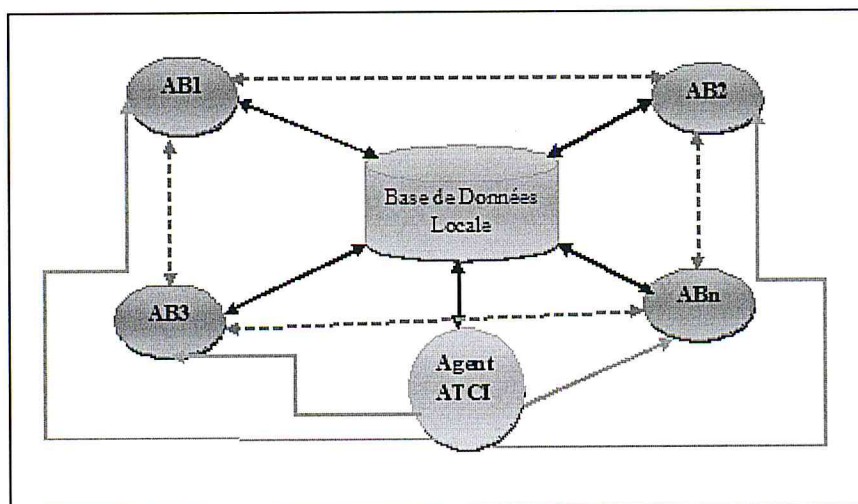


Fig.VI.2. L'organisation du système.

5. Conception détaillée

Dans cette partie, qui est certainement l'une des plus importantes de notre projet, nous allons établir la structure opérationnelle des composants de notre système multi-agents. Dans ce qui suit, nous décrivons :

- Le diagramme de cas d'utilisation.

- L'architecture interne des agents, c'est-à-dire qu'ils soient cognitifs ou réactifs.
- La représentation des agents sous forme de classe AUML qui détaillera entre autres les actions et les méthodes.
- Pour les agents cognitifs, définir leur module d'apprentissage et de raisonnement à base de Système de Classeurs.
- Description des diagrammes d'activités des agents.
- Conception de la base de données utilisée par l'application.

5.1. Diagramme de cas d'utilisation

Ce diagramme montre tous les cas d'utilisation possibles reliés chacun à son propre acteur ainsi que les dépendances entre ces cas.

Nous distinguons deux types d'agent :

Agent ATCI

Il réalise les fonctions suivantes :

- Déclenchement ouverture journée compensation.
- Alimenter BDD locale qui inclut l'Extraction de données de la base de données système.
- Déclencher fermeture journée compensation qui inclut la génération d'un fichier XML et la fermeture des séances.

Agent Banque :

Il réalise les fonctions suivantes :

- Lecture des Opérations de BDD locale.
- Préparation paramètres CS (Etat Banque, Type Banque) qui inclut le Calcul des Paramètres (Taux d'Evolution Mensuel, Taux d'Evolution Annuel, Taux de Rejet, Ratio).
- Déclencher Cycle CS (Classifier System) qui inclut le Calcul Reward et prendre la décision.
- Calculer Solde à Chaque Opération.
- Mise à jour état transitoire BDD locale.

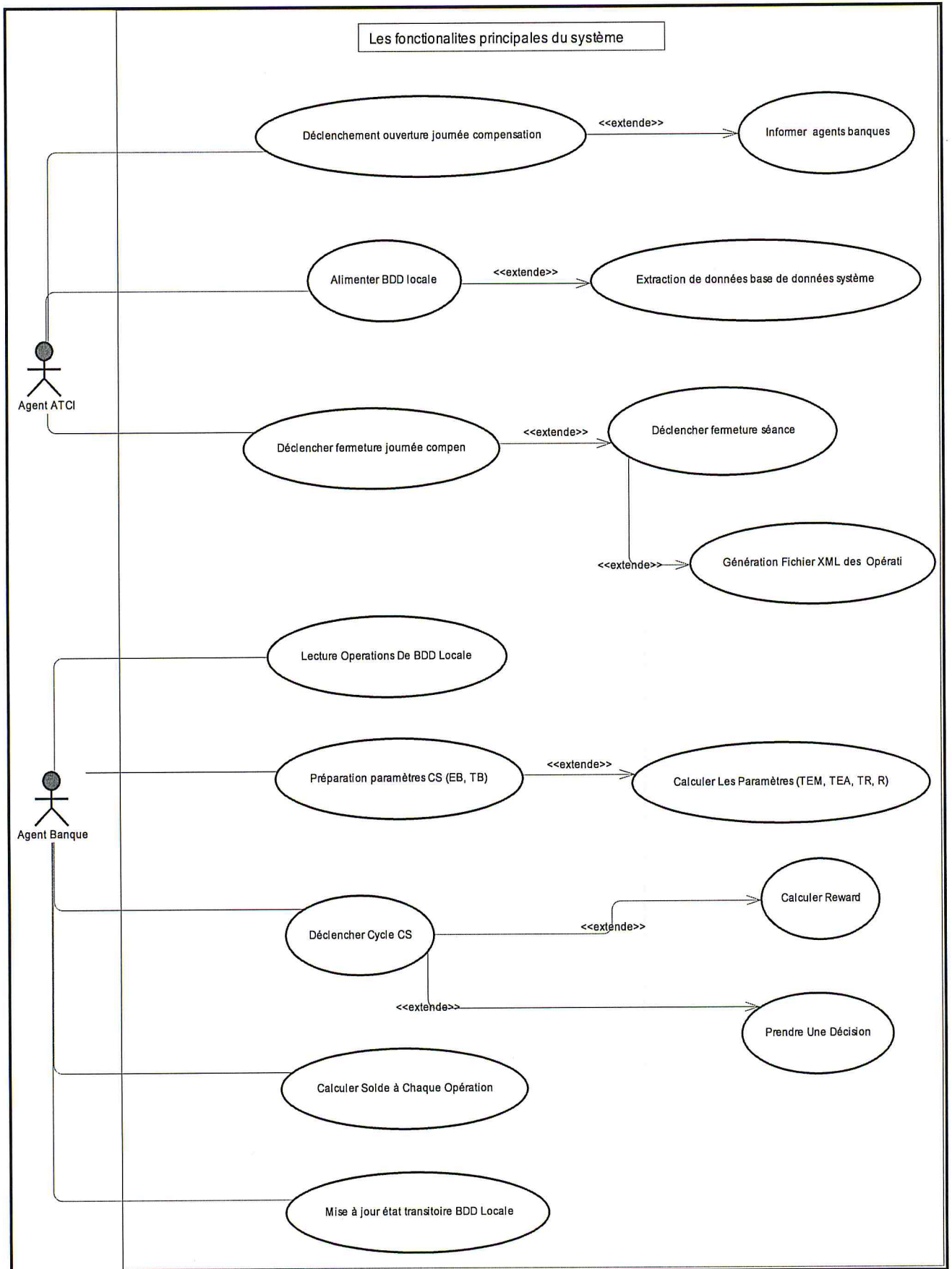


Fig.VI.3 Diagramme de cas d'utilisation (UML).

5.2.Agent ATCI

5.2.1. Architecture et environnement de l'agent ATCI

L'agent ATCI se compose de 5 modules principaux :

- Un module de déclenchement d'ouverture d'une journée de compensation qui sert à l'initialisation des agents banques.
- Un module d'extraction de données depuis la base de données système pour le traitement des opérations.
- Un module d'alimentation de la base de données locale, qui permet aux agents banques par la suite le recueil des opérations à traiter.
- Un module de déclenchement de la fermeture d'une journée de compensation qui sert à mettre fin aux traitements des agents banques.
- Un module responsable de la génération d'un fichier XML.

L'agent ATCI gère deux bases de données, la première est celle du système qui est la source principale de toutes les opérations échangées entre les banques ; il l'utilise pour alimenter la base de données locale qui est exploitée par les agents banques pour le traitement des opérations.

L'Agent ATCI est un agent réactif, il réagit à chaque fois que le système reçoit un ordre de paiement soumis par une Banque. Son architecture interne est représentée comme suit :

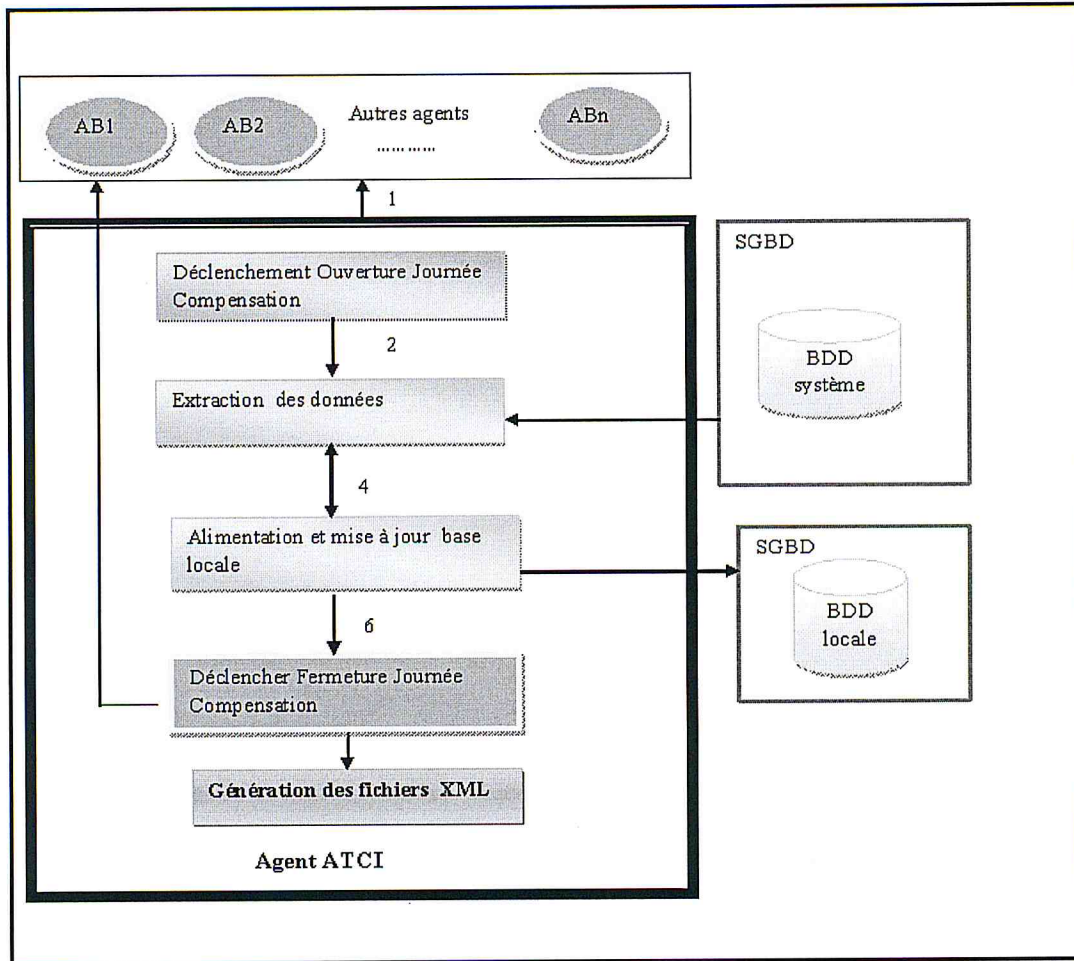


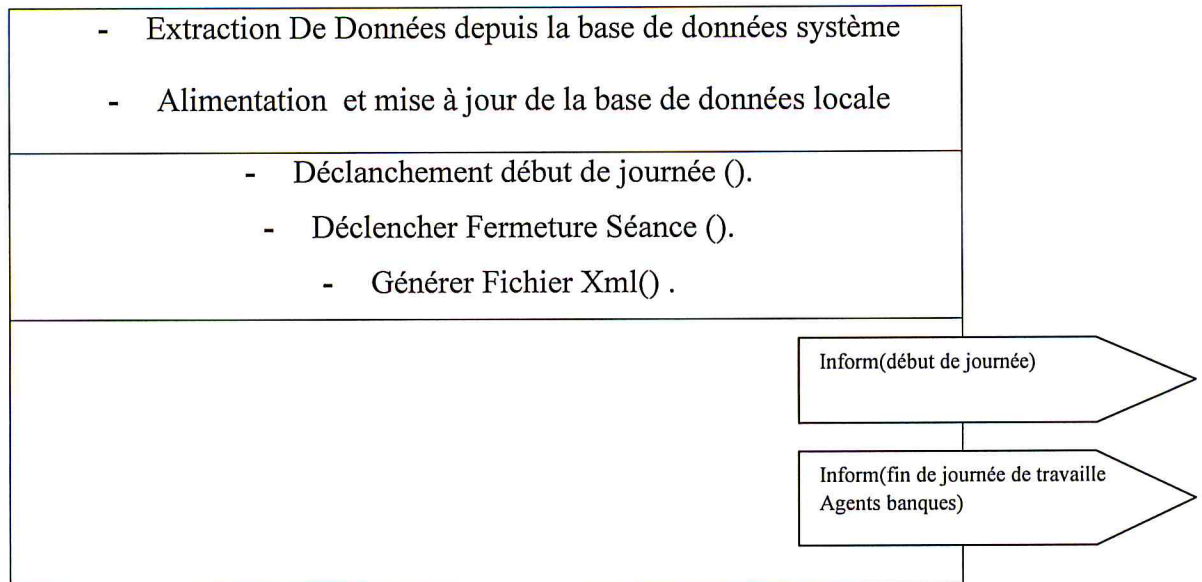
Fig.VI.4 Architecture interne de l'agent ATCI

5.2.2. Classe d'Agent ATCI

Cette description nous amène à bien comprendre la structure de la classe Agent ATCI. Nous définissons d'une manière globale la classe de l'agent ATCI tel que nous avons quatre parties :

- La première comporte les différents attributs.
- La deuxième comporte les fonctions globale de l'agent ATCI
- La troisième partie comporte les fonctions élémentaires.
- La dernière partie comporte les interactions effectuées par l'agent ATCI

Classe Agent ATCI	
-	Liste des codes des Agents Banques
-	ID_OPERATION, ID_BANQUE_DEST, ID_BANQUE_REME



Tab.VI.1 Représentation de la classe de l'agent ATCI.

5.2.3. Description de l'Agent ATCI

- **Les Actions**

- **Extraction De Données depuis la base de données système ()** : permet la récupération des données depuis la base de données système.
- **Alimentation et mise à jour de la base de données locale ()** : permet l'insertion des opérations dans la base de données locale.

- **Les méthodes**

Déclanchement début de journée () : permet aux agents de commencer leurs travaille.

Déclencher Fermeture Séance () : permet aux agents de mettre fin à leur travail journalier.

Générer Fichier Xml() : permet de regrouper toute les opération traiter dans une séance de compensation sous format XML.

5.3.Agent Banque (AB) :

Un agent banque est un agent cognitif doté de buts. Son but principal est de prendre des décisions qui sont optimal. Pour y arriver, l'AB doit apprendre intelligemment afin d'améliorer son comportement vers les opérations de paiement traitées et ainsi sa rentabilité (utilité). Pour atteindre ses buts il devra

être muni d'un module d'apprentissage qui lui permettra de raisonner et de choisir les bonnes actions.

5.3.1. Module de raisonnement et d'apprentissage

C'est le module qui permet à l'agent de s'adapter à son environnement et d'apprendre de ses expériences passées pour prendre des décisions sur les actions à exécuter. Dans notre conception, nous avons opté pour les systèmes de classeurs pour réaliser cet apprentissage.

Comme nous l'avons vu dans le chapitre des systèmes de classeurs, ces derniers permettent aux agents un apprentissage par renforcement. Un système de classeurs est en pratique un ensemble de règles qui déterminent la politique de comportement d'un agent. Il possède un mécanisme d'évaluation de ses règles en rémunérant celles qui produisent plus de gain, donc qui permettent une meilleure adaptation à l'environnement, et en éliminant celles qui en produisent peu. Le système démarre avec un certain nombre de règles prédéfinies, d'autres sont générées au fur et à mesure du temps selon les situations présentes dans l'environnement, ceci donne un avantage à l'agent de construire d'autres règles de comportement par son propre apprentissage.

Cette technique d'apprentissage convient à notre problème car les agents banques, à leur naissance, essaient de s'adapter dans un environnement qu'ils ne maîtrisent pas encore en évaluant périodiquement chacune des actions qu'ils choisissent.

Avant de décrire le module d'apprentissage des agents Banques, nous admettons ce qui suit :

- Tous les agents banques ont le même module d'apprentissage constitué d'un seul système de classeurs:
- Le système de classeurs permet à chaque agent d'apprendre de ses propres actions. Il sert à choisir la bonne action que l'agent doit prendre en se basant sur à la fois, l'état banque, le type d'instrument à traiter, la limite banque, type banque ; et d'autres paramètres calculés comme taux de rejet, taux d'évolution mensuel...etc.
- Chaque agent banque connaît tous les autres agents banques.
- L'espace des actions à choisir est le même pour tous les agents, et chaque agent sait que les autres choisissent leurs actions dans cet espace.

Maintenant que les hypothèses de départ sont posées, nous allons détailler les règles constituant le système de classeurs des agents en détaillant les conditions, les actions ainsi que la fonction d'évaluation qui permet d'évaluer les règles et d'en sélectionner les meilleures.

5.3.1.1. Système de classeurs

Ce système de classeurs définit la politique de choix de l'action appliquée par l'agent. Le système évalue l'état de banque, type d'opération, type banque, limite banque ainsi que les paramètres calculable afin de choisir la meilleure décision qui réduit toutes chance de tomber dans une situation de défaillance

a. La structure du système de classeurs de l'agent banque

Structure de la règle

La règle est constituée de trois parties. La première partie est la partie condition, elle est représentée sur 21 bits. La deuxième partie est la partie action (décision), elle est représentée sur 2 bits, et la dernière partie est la partie fitness qui contient la force de la règle.

Partie « Condition » ; elle est constituée de 2 parties:

Une partie Cas ; contient tous les cas possibles rencontrés par une banque durant une séance de compensation.

Une partie Paramètres ; contient 7 paramètres qui contribuent à la validation de la décision prise dans la partie Cas.

Partie « Action » ; elle représente l'action que doit réaliser l'agent. On distingue 4 possibilités :

- **Accepter** : cette opération est réalisable.
- **Retarder** : cette opération est retardée à la séance prochaine.
- **Rejeter** : cette opération est rejetée pour cette journée mais peut être traitée dans la journée suivante.
- **Annuler** : cette opération est annulée totalement du système, avec possibilité de traitement dans la journée suivante.

Partie fitness ; contient la force de la règle et est un réel compris entre [0,1].

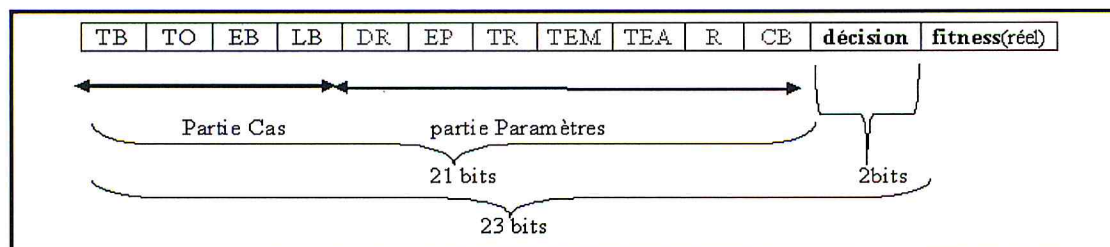


Fig.VI.5 Représentation d'une règle du système de classeurs.

b. Les paramètres et leur codification

b.1. Les paramètres de la partie cas

TB : Type de Banque ; d'après la position de la banque par rapport aux moyens de paiement on a 2 possibilités soit qu'elle est un émetteur ou bien un récepteur de l'instrument.

C'est-à-dire Remettante ou Destinataire. Sa codification est faite sur 1 bit.

Type des banques	Codification binaire	Codification numérique
Remettante	0	0
Destinataire	1	1

Tab VI.2 Codification binaire « Type Banque ».

TO : Type Opération ; nous avons vu dans le chapitre I « les systèmes de paiement de masse » les différents moyens de paiement. Ils sont codifiés sur 4 bits :

L'instrument de paiement	Codification binaire	Codification numérique
Chèque accompagné de sa valeur papier.	0000	0
Chèque accompagné de son image scannée.	0001	1
chèque électronique (pas de papier ni d'image).	0010	2
Ordre de prélèvement	0011	3
Retrait sur DAB/GAB.	0100	4

Paiement sur TPE	0101	5
Les lettres de change	0110	6
Les billets à ordre.	0111	7
Virement standard de clientèle.	1000	8
Virement de banque à banque.	1001	9
Virement de régularisation	1010	10

Tab VI.3 Codification binaire « Type Opération ».

EB : Etat Banque ; au démarrage de notre système toutes les banques ont le même état 'Créditrice' mais après la 1^{ère} séance chaque banque aura son propre état en fonction de ses échanges interbancaires.

L'état de la banque est obtenu par la formule suivante : **EB = C-D.**

Tels que : **C = ∑ des moyens de paiements à crédit(en capital).**

D= ∑ des moyens de paiements à débit(en capital).

Nous avons deux états différents qui sont codifiés sur 1 bit.

Si **C-D** est un nombre positif alors la banque est Créditrice : 0 sinon elle est Débitrice : 1.

Etat banque	Codification binaire	Codification numérique
Créditrice	0	0
Débitrice	1	1

Tab VI.4 Codification binaire « Etat Banque ».

LB : Limite Banque ; est calculée par la formule suivante : **LB= (C-D)/X.**

Tels que : **C = ∑ des moyens de paiements à crédit.**

D= ∑ des moyens de paiements à débit.

X=80% du budget initial de chaque banque.

Nous nous intéressons seulement aux cas où le rapport **(C-D)/X** est négatif, c'est-à-dire les banques débitrices.

Pour cela on a proposé des intervalles pour ce rapport qui sont codifiés sur 2 bits, tels que :

L'intervalle	Degré	La signification	La codification binaire	Codification numérique
[0, 0.4 [moins grave	On retarde les opérations à débit.	00	0
[0.4, 0.7 [un peut grave	On fait un tri sur les opérations à accepter	01	1
[0.7, 0.9 [Grave	On envoi des alarmes au participant et on fait des tris sur les opérations à accepter	10	2
>1	très grave	On annule les opérations à débit et on fait des tris sur les opérations à accepter	11	3

Tab VI.5 Codification binaire « Limite Banque ».

b.2. La codification des paramètres de la deuxième partie du SC

DR: Date de Règlement ; c'est la date limite pour régler chaque type de moyen de paiement. Elle est codifiée sur 2 bits :

Date de règlement	Codification binaire	Codification numérique
J	00	0
j+1	01	1
J+2	10	2

Tab VI.6 Codification binaire « Date de Règlement ».

EP: Etat de la journée Précédente ; C'est l'état final de la journée précédente après règlement du solde, il est codifié sur 1 bit :

Etat de la journée précédente	Codification binaire	Codification numérique
Créditrice	0	0
Débitrice	1	1

Tab VI.7 Codification binaire « Etat de la journée Précédente ».

TR: Taux de Rejet ; C'est le taux des opérations rejetées par rapport au nombre total des opérations. Il est calculé par la formule suivante :

$$\text{Taux de Rejet} = (\text{rejet} / (\text{traité} - \text{rejet})).$$

Tel que **rejet** : nombre des opérations rejetées.

Traité : nombre de toutes les opérations traitées.

Le Taux de Rejet est codifié sur 2 bits.

Taux de rejet	Codification binaire	Codification numérique
[0, 0.4 [00	0
[0.4, 0.7 [01	1
[0.7, 1[10	2

Tab VI.8 Codification binaire « Taux de Rejet ».

TEM: Taux d'Evolution par Mois ; c'est une comparaison des montants (le capital) entre 2 mois de la même année pour savoir le taux d'évolution.

Nous le calculons par la formule suivante :

$$\text{Taux d'Evolution par Mois} = ((N-M) / M) * 100.$$

Tel que : **N** : le montant du mois courant.

M : le montant du mois précédent.

Il est codifié sur 2bits.

Taux d'évolution par mois	Codification binaire	Codification numérique
<0	00	0

[0,50[01	1
[50,100[10	2
>100	11	3

Tab VI.9 Codification binaire « Taux d'Evolution par Mois ».

TEA: Taux d'Evolution par Année ; c'est une comparaison des montants (le capital) entre le mois de l'année courante et le même mois de l'année précédente pour savoir le taux d'évolution.

Nous le calculons par la formule suivante :

$$\text{Taux d'Evolution par Année} = (N / (N-1)) * 100.$$

Tel que : **N** : le montant total de l'ensemble des opérations traitées du mois de l'année courante.

N-1 : le montant total de l'ensemble des opérations traitées du mois de l'année précédente.

Il est codifié sur 2bits.

Taux d'évolution par année	Codification binaire	Codification numérique
<0	00	0
[0,50[01	1
[50,100[10	2
>100	11	3

Tab VI.10 Codification binaire « Taux d'Evolution par Année ».

R: Ratio ; c'est un paramètre qui nous aide à savoir la position d'un instrument de paiement par rapport à l'ensemble des moyens de paiements. Elle est calculée par la formule suivante : **R= A/ T.**

Tel que : **A** : c'est l'instrument qu'on veut savoir sa position.

T : c'est le montant total des instruments traitées qui utilise une banque donnée.

Elle est codifiée sur 2 bits :

Ratio	Codification binaire	Codification numérique
[0, 0.4 [00	0
[0.4, 0.7 [01	1
[0.7, 1]	10	2

Tab VI.11 Codification binaire « Ratio ».

CB: Classe Banque ; C'est la catégorie de la banque elle est codifié sur 2 bits

Classe banque	Codification binaire	Codification numérique
Grande	00	0
Moyenne	01	1
Petite	10	2

Tab VI.12 Codification binaire « Classe Banque ».

c. ACTION ; elle est codifiée sur 2 bits

Action	Codification binaire	Codification numérique
Accepter	00	0
Retarder	01	1
Rejeter	10	2
Annuler	11	3

Tab VI.13 Codification binaire « ACTION ».

d. Les valeurs des paramètres

Les paramètres	Description	Valeurs
DR	Date de règlement	1
		2
		3
EP	Etat précédent	C : 2 D : 1

TR	Taux de rejet	[0, 1]
TEM	Taux d'évolution par mois	[-1,4]
TEA	Taux d'évolution par année	[-1,4]
CB	Classe banque	Grande : 3 Moyenne : 2 Petite : 1
R	Ratio	[0, 1]

Tab VI.14 Valeurs des paramètres.

e. Les poids des paramètres

Si action « Accepter »

Les paramètres	Le poids	Signification du poids
DR	3	Important
EP	2	modérément Important
TR	2	modérément important
TEM	2	modérément important
TEA	2	modérément important
CB	3	Important
R	1	Peu important

Tab VI.15 Les poids des paramètres « action Accepter ».

Si action « Retarder »

Les paramètres	Le poids	Signification du poids
DR	2	modérément important
EP	2	modérément important
TR	2	modérément important
TEM	2	modérément important

TEA	2	modérément important
CB	3	Important
R	2	modérément important

Tab VI.16 Les poids des paramètres « action Retarder ».

Si action « Rejeter »

Les paramètres	Le poids	Signification du poids
DR	1	peu important
EP	2	modérément important
TR	4	Très important
TEM	2	modérément important
TEA	2	modérément important
CB	3	Important
R	1	peu important

Tab VI. 17. Les poids des paramètres « action Rejeter ».

Si action « Annuler »

Les paramètres	Le poids	Signification du poids
DR	1	peu important
EP	1	Peu important
TR	4	Très important
TEM	2	modérément important
TEA	2	modérément important
CB	3	Important
R	2	modérément important

Tab VI.18 Les poids des paramètres « action Annuler ».

f. Les cas possibles

- 1) Si (TB=R et TO= (carte, prélèvement, cheque, effet)) ou (TB=D et TO=virement) alors **accepter**.
- 2) Si (TB= R et TO= virement et EB=C) ou (TB=D et TO = (carte, prélèvement, cheque, effet) et EB=C) alors **accepter**.
- 3) Si TB=R et TO =virement et EB=D et $LB \in [0,0.4 [)$ ou (TB=D et TO= (carte, prélèvement, cheque, effet) et EB=D et $LB \in [0, 0,4[)$ alors **retarder**.
- 4) Si (TB =R et TO= virement et EB=D et $LB \in [0.4, 0.7 [)$ ou (TB=D et TO= (carte, prélèvement, cheque, effet) et EB=D et $LB \in [0.4, 0.7 [)$ alors **rejeter**.
- 5) Si (TB =R et TO= virement et EB=D et ($LB \in [0.7, 1[$ ou $LB > 1$)) ou (TB=D et TO= (carte, prélèvement, cheque, effet) et EB =D et ($LB \in [0.7, 1[$ ou $LB > 1$)) Alors **annuler**.

g. Évaluation des règles: la fonction reward

Comme nous avons dit dans la description de classeur la partie condition est divisée en deux, une partie Cas qui est la plus importante et qui influe beaucoup sur la décision finale, elle a un poids (coefficient) de 5/8 de la décision. La deuxième partie aide la première partie pour voir si on valide le cas ou non, elle influe par un petit coefficient par rapport à la 1^{ère} partie et elle prend 3/8.

On distingue deux cas dans le reward : des cas où on va récompenser une décision c.-à-d. que c'est la bonne décision et un cas où la décision est moins rétribuée et pour cela nous avons introduit le petit nombre 1/4 qu'on va le multiplier par 5/8 dans ce cas.

Remarque : RE= reward.

Remarque : Pour borner la valeur du reward dans l'intervalle [0, 1] alors il faut diviser la valeur trouvée par 2.

Si action= accepter

Si (TB=R et TO= (carte ou prélèvement ou cheque ou effet)) ou (TB=D et TO=virement)

Alors

$$RE = (5/8 * 1 + 3/8 * [(3 * DR) + (2 * EP) + (2 * TR) + (2 * TEM) + (2 * TEA) + (3 * CB) + (1 * R)] / \sum \text{des poids}) / 2.$$

Si (TB = R et TO = virement) et EB = C) ou (TB = D et TO = (carte, prélèvement, chèque, effet) et EB = C)

Alors

$$RE = (5/8 * 1 + 3/8 * [(3 * DR) + (2 * EP) + (2 * TR) + (2 * TEM) + (2 * TEA) + (3 * CB) + (1 * R)] / \sum \text{des poids}) / 2.$$

Sinon

$$RE = ((5/8 * 1/4) + 3/8 * [(3 * DR) + (2 * EP) + (2 * TR) + (2 * TEM) + (2 * TEA) + (3 * CB) + (1 * R)] / \sum \text{des poids}) / 2.$$

Si action = retarder

Si TB = R et TO = virement et EB = D et LB ∈ [0, 0.4 [) ou (TB = D et TO = (carte, prélèvement, chèque, effet) et EB = D et LB ∈ [0, 0,4 [)

Alors

$$RE = (5/8 + 3/8 * [(2 * DR) + (2 * EP) + (2 * TR) + (2 * TEM) + (2 * TEA) + (3 * CB) + (2 * R)] / \sum \text{des poids}) / 2.$$

Si non

$$RE = ((5/8 * 1/4) + 3/8 * [(2 * DR) + (2 * EP) + (2 * TR) + (2 * TEM) + (2 * TEA) + (3 * CB) + (2 * R)] / \sum \text{des poids}) / 2.$$

Si action = rejeter

Si (TB = R et TO = virement et EB = D et LB ∈ [0.4, 0.7 [) ou (TB = D et TO = (carte, prélèvement, chèque, effet) et EB = D et LB ∈ [0,4,0,7 [).

Alors !

$$RE = (5/8 + 3/8 * [(1 * DR) + (2 * EP) + (4 * TR) + (2 * TEM) + (2 * TEA) + (3 * CB) + (1 * R)] / \sum \text{des poids}) / 2.$$

Si non

$$RE = ((5/8 * 1/4) + 3/8 * [(1 * DR) + (2 * EP) + (4 * TR) + (2 * TEM) + (2 * TEA) + (3 * CB) + (1 * R)] / \sum \text{des poids}) / 2.$$

Si action = annuler

Si (TB = R et TO = virement et EB = D et (LB ∈ [0.7, 1[ou LB > 1)) ou (TB = D et TO = (carte, prélèvement, chèque, effet) et EB = D et (LB ∈ [0.7, 1[ou LB > 1)) .

Alors

$$RE = \frac{5}{8} + \frac{3}{8} * \left(\frac{(1*DR) + (1*EP) + (1*TR) + (2*TEM) + (2*TEA) + (3*CB) + (2*R)}{\sum \text{des poids}} \right) / 2.$$

Si non

$$RE = \frac{((5/8 * 1/4) + 3/8 * \left(\frac{(1*DR) + (1*EP) + (1*TR) + (2*TEM) + (2*TEA) + (3*CB) + (2*R)}{\sum \text{des poids}} \right))}{2}.$$

5.3.2. Architecture d'un agent Banque :

L'architecture d'un agent Banque se compose de 10 modules :

- Un module d'envoi d'information (pour la mise à jour du solde de l'agent banque représentant la banque destinataire).
- Un module de réception d'information (pour la mise à jour du solde de l'agent banque représentant la banque remettante).
- Un module de lecture des opérations.
- Un module responsable de la préparation et du calcul des paramètres CS.
- Un module de déclenchement d'un cycle CS.
- Un module responsable du calcul de solde.
- Un module chargé de prendre une décision transitoire pour chaque opération.
- Un module d'apprentissage constitué d'un système de classeurs qui lui permet de raisonner pour faire des choix sur les décisions à prendre.
- Un module de mise à jour de l'état transitoire de la base de données locale.

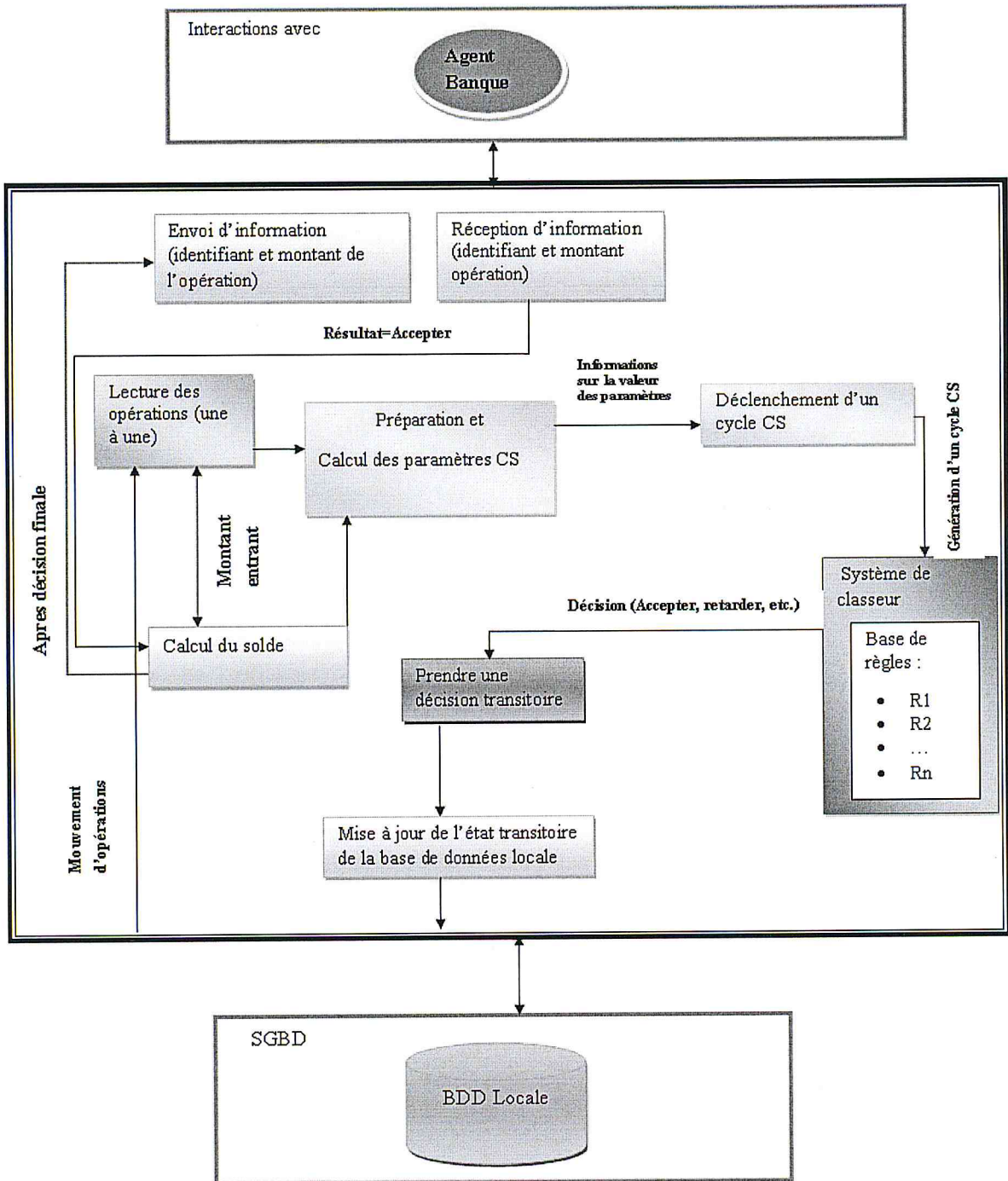


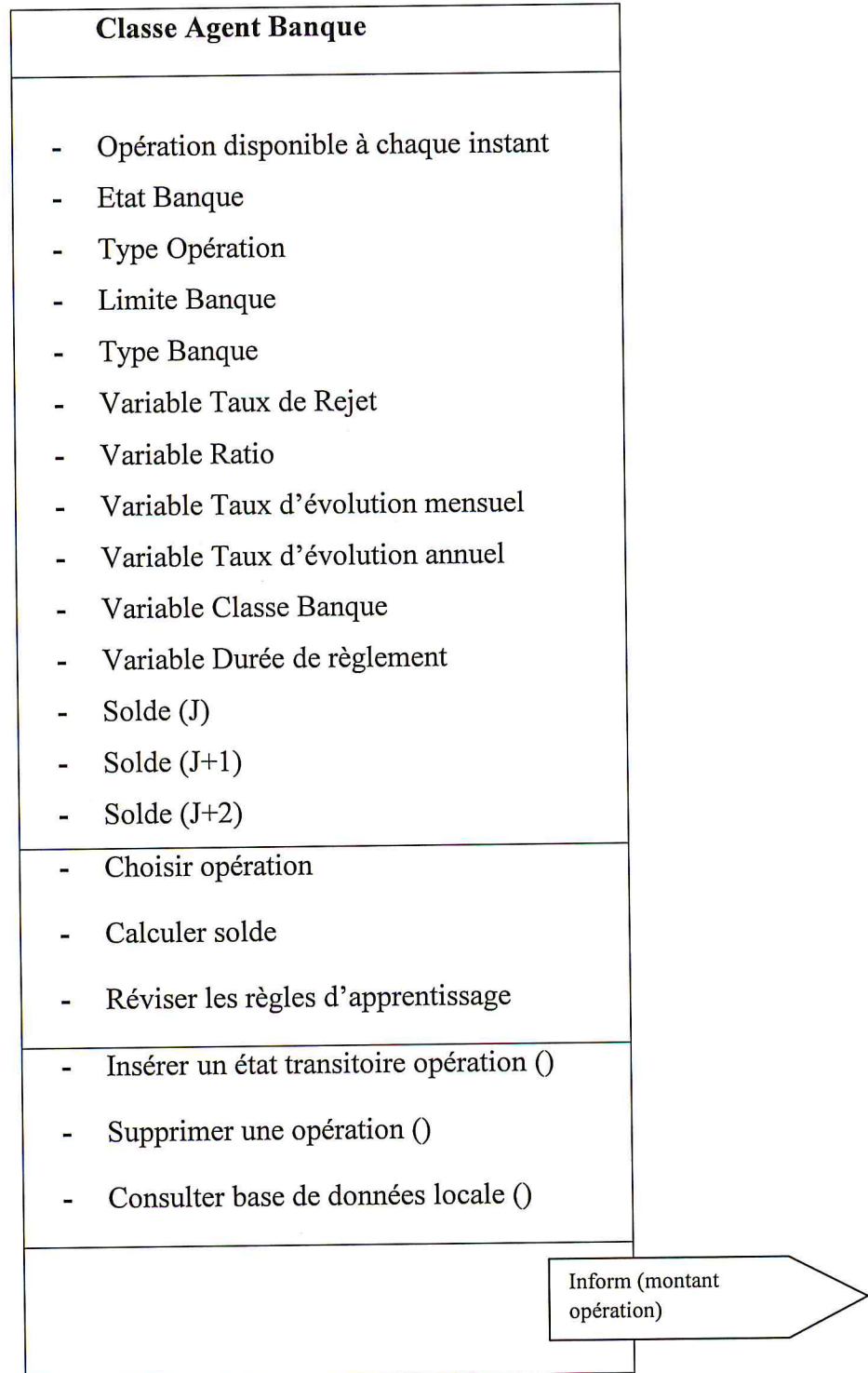
Fig.VI.6 Architecture interne de l'agent Banque.

5.3.3. Classe d'Agent Banque :

Cette description nous amène à bien comprendre la structure de la classe Agent Banque :

Nous définissons d'une manière globale la classe de l'agent Banque tel que nous avons quatre parties :

- La première comporte les différents attributs.
- La deuxième comporte les fonctions globale de l'agent Banque.
- La troisième partie comporte les fonctions élémentaires.
- La dernière partie comporte les interactions effectuées par l'agent Banque.



Tab VI.19 Représentation de la classe de l'agent banque.

5.3.4. Description de l'Agent Banque

Les Actions

- **Choisir opération :** Permet à l'Agent Banque de Choisir l'opération à traiter.
- **Calcule du solde :** Permet à l'Agent Banque de calculer le solde en temps réel.
- **Réviser les règles d'apprentissage :** Permet à l'Agent Banque de réviser les règles d'apprentissage, en gardant les meilleures règles.

Les méthodes

Insérer un état transitoire opération () : Permet de mettre à jours l'état transitoire après traitement de l'opération.

Supprimer une opération () : Permet de supprimer une opération de la base de données local.

Consulter base de données local () : permet de savoir l'état définitif de chaque opération.

5.4.Processus global des traitements des opérations de paiement

Ce processus se décompose en trois séances de compensation

Première séance de compensation

Cette séance est synonyme de début de journée. Elle se caractérise par la reprise et traitement des opérations déjà insérées dans la base de données locale ; ces opérations représentent des échanges interbancaires établis dans des journées antérieures (précédentes : J-1, J-2, J-3, etc.). Les raisons pour laquelle ces opérations figurent toujours dans notre BDD locale est :

- Etat transitoire des opérations est égal à : retarder, rejeter, annuler.
- Des opérations qui ont un état transitoire est égal à accepter, mais la réception de confirmation d'acceptation ou de rejet bancaire de la part de la banque destinataire n'a pas eu lieu.
- Des opérations qui ont un état définitif est égal à accepter, mais leurs date de règlement n'est pas encore atteinte ; à noter que la date de règlement se diffère selon le type d'instrument (J, J+1, ou bien J+2).

Cette première séance prend en charge et traite aussi toutes les **nouvelles** opérations arrivées en jour J.

L'opération peut subir l'un des deux traitements suivant

1) Opération déjà traitée par notre système, c'est-à-dire des opérations qui disposent d'un état transitoire (retarder, rejeter, annuler, accepter). Notre système prend en charge ces opérations et génère un cycle CS pour chacune, afin d'aboutir à un nouvel état. Les opérations qui ont comme état transitoire « accepter » ne vont pas être reprises, car c'est l'état optimal.

2) Opérations non traitées, qui sont tous concernées par la génération d'un cycle CS (le cas des nouvelles opérations).

Deuxième séance de compensation

Contrairement à la première séance, cette séance reprend uniquement les opérations qui ont un état transitoire égal à « retarder » ainsi que les opérations qui arrivent au cours de cette séance.

Les opérations « Annuler » et « Rejeter » vont être reprises la journée suivante pour leur données plus de chance d'être réglées.

Troisième séance de compensation

Elle est identique à la deuxième séance dans son traitement sauf qu'avant la clôture de cette séance, un processus de mise à jour de la base de données locale se déclenche et établit les modifications nécessaires.

A noter que toutes les séances ont comme point commun la génération d'un OUTGO (format XML) à la fin de chaque séance.

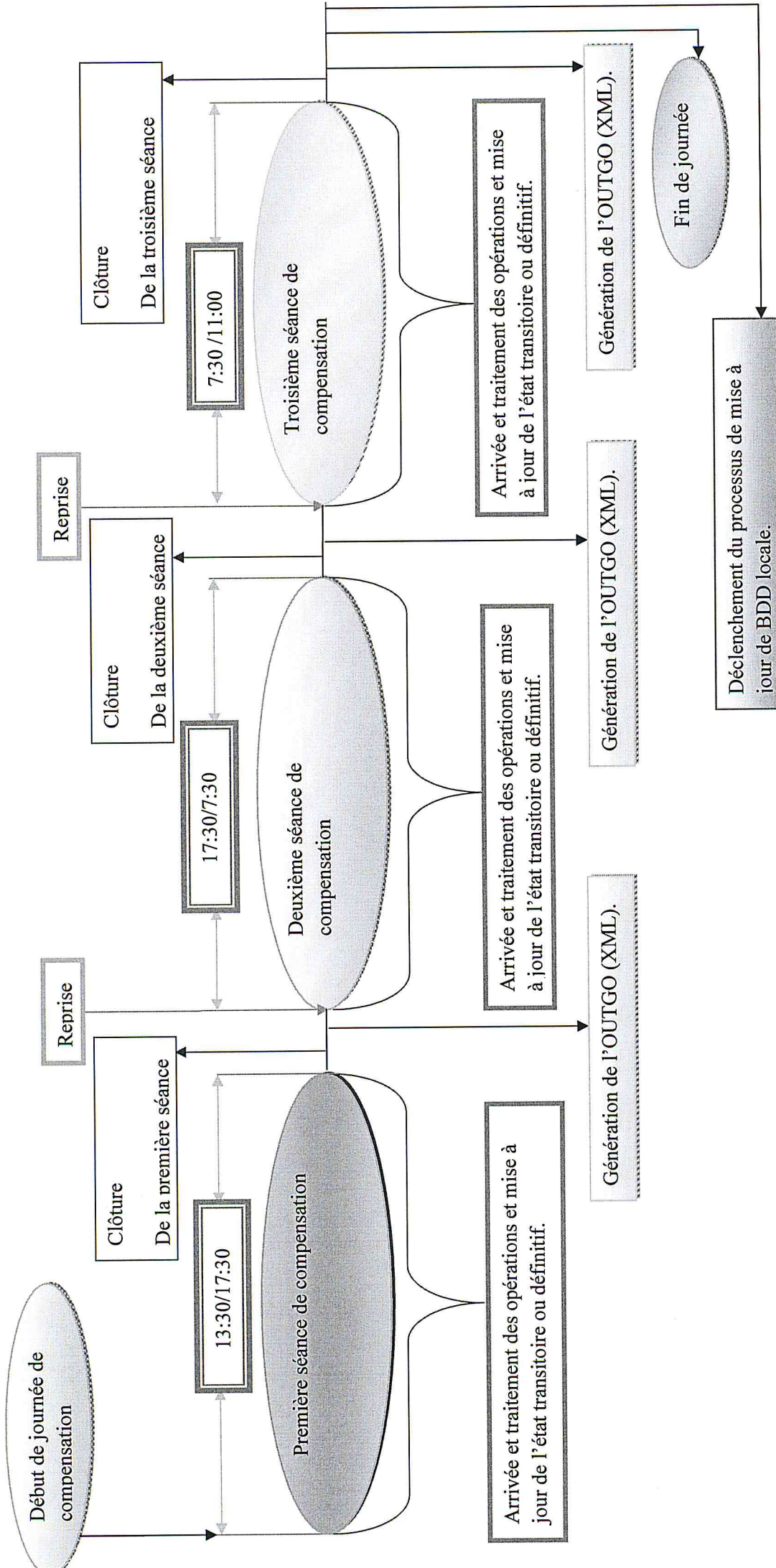


Fig.VI.7 Enchaînement des traitements durant une journée de compensation (3 séances)

5.5. Les diagrammes d'activités

5.5.1. Diagramme d'activité de l'ouverture d'une journée de compensation

Ce diagramme nous démontre le début d'une journée de compensation qui est déclenchée par l'agent ATCI ;

Elle est suivie par l'initialisation des agents banque.

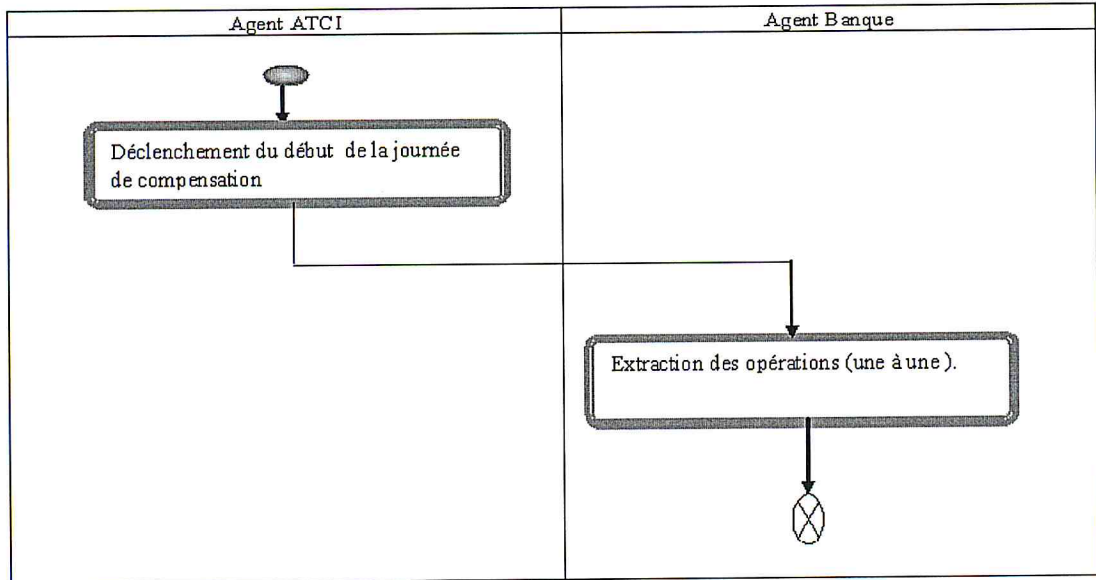


Fig.VI.8 Diagramme d'activité de l'ouverture d'une journée de compensation (UML).

5.5.2. Diagramme d'activité de la fermeture d'une journée de compensation

Ce diagramme exprime la fin d'une journée de compensation ; déclenchée souvent par l'agent ATCI qui est responsable d'achèvement de la journée de compensation. A noter que l'agent ATCI envoie des messages aux agents banques afin de les prévenir.

Le traitement consiste à faire des mises à jours de la part de l'agent banque.

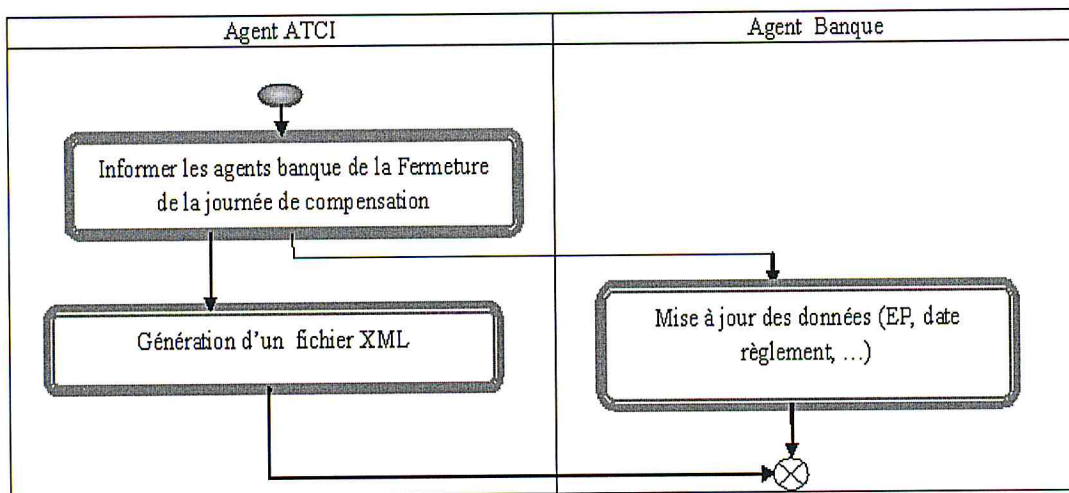


Fig.VI.9 Diagramme d'activité de la fermeture d'une journée de compensation (UML).

5.5.3. Diagramme d'activité du traitement appliqué sur une opération et prise de décision

Ce diagramme illustre le traitement subit par une opération après sa réception. Le processus se déclenche dès la réception d'une remise aller. L'agent ATCI fait l'extraction de toutes les opérations et met à jour notre base de données (locale). Ensuite l'agent banque pointe sur l'opération et la traite, si elle le concerne (C'est-à-dire, il faut que le nom de la banque qu'il représente figure dans l'opération).

Le traitement d'une opération peut suivre l'un des deux chemins suivant :

Premier chemin

Si l'opération n'a pas été traité ; alors notre système va l'évaluer et aboutit vers la fin à une décision qui est transitoire. Cet état doit être pris en considération, donc la base de données (locale) va subir une mise à jour, en attendant l'état définitif transmis par la banque en cas où, état transitoire est égal à 'accepter' ; mais dans les autres cas, c'est-à-dire : retarder, rejeter ou bien annuler, la réponse de la banque n'est pas nécessaire.

Deuxième chemin

L'opération est déjà traitée, mais ce qui est important dans ce cas ; c'est la décision finale de la banque. Si la banque ne rejette pas l'opération .Alors l'opération arrive à sa fin et aucun changement n'aura lieu. Sinon, l'opération va être retirée et le solde va être recalculé. Pour cela on distingue 4 cas :

Type banque est symbolisé par TB.

Type opération est symbolisé par TO.

TB=R et TO=chèque, carte, prélèvement, effet. (Solde - montant opération).

TB=R et TO=virement. (Solde + montant opération).

TB=D et TO=cheque, carte, prélèvement, effet. (Solde + montant opération).

TB=D et TO=virement. (Solde - montant opération).

Sans oublier la mise à jour de l'état définitif de la base. Dans tous les cas le traitement finit toujours par la génération d'un fichier XML à la fin de chaque séance de compensation.

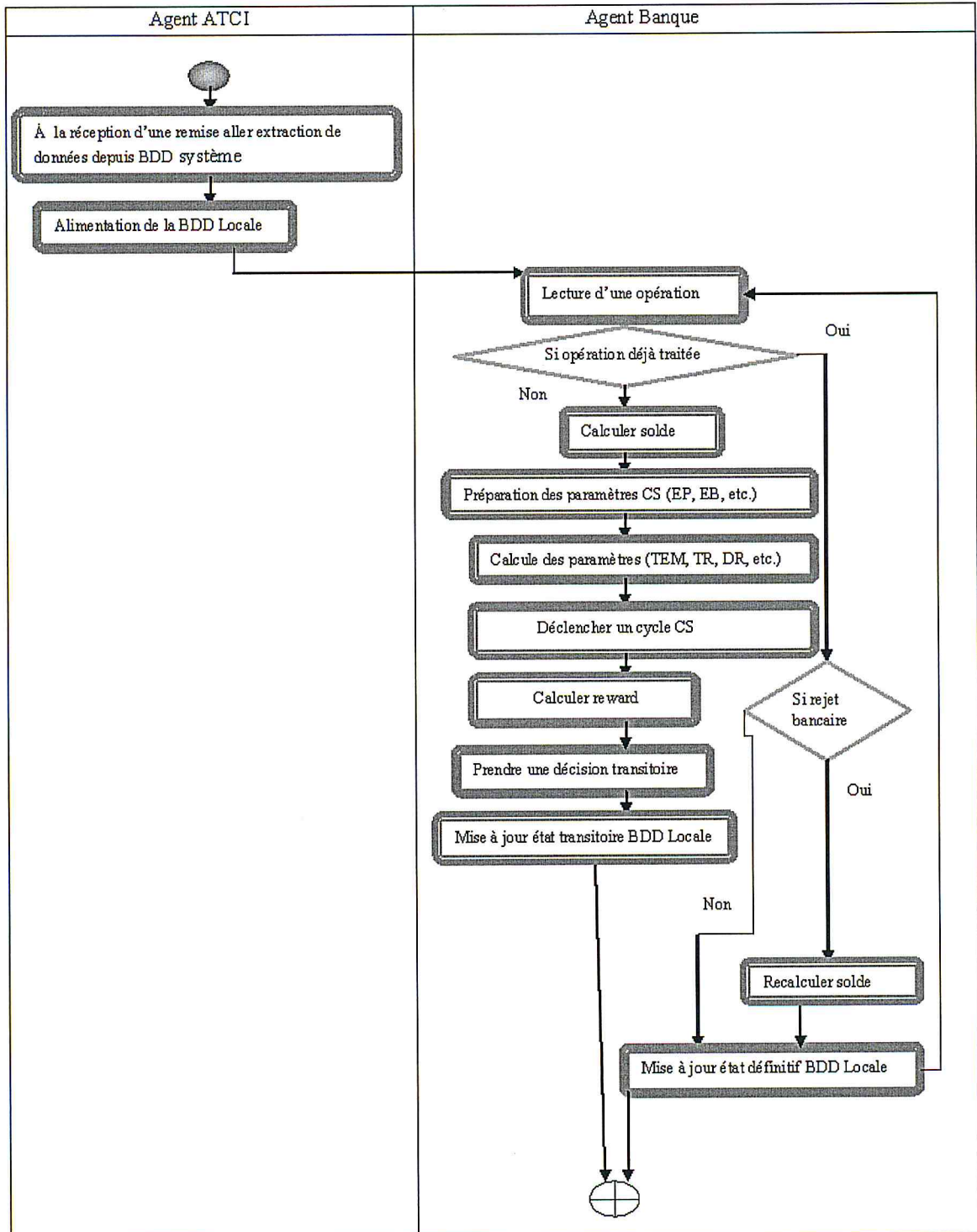


Fig. VI.10 Diagramme d'activité du traitement appliqué sur une opération et prise de décision (UML).

5.5.4. Diagramme d'activité des traitements de l'agent banque

Ce diagramme donne une vue un peu détaillé du traitement appliqué sur les opérations, et surtout démontrer la stratégie conçue pour gérer n'importe qu'elle genre de conflit. N'oublions pas que notre objectif est d'éloigner toute probabilité de tomber dans une situation de défaillance, donc les opérations comme virement ont un résultat négatif plus sur les banques remettantes que destinataires. On a donc donné la priorité de décision aux banques remettantes. Et c'est le même principe pour les autres instruments de paiement: chèques, effets, carte, prélèvement où on donne priorité aux banques destinataire. A noter qu'après le traitement déjà expliqué dans le diagramme d'activité précédent, l'agent banque doit informer son confrère si l'état =accepter pour la mise à jours du solde.

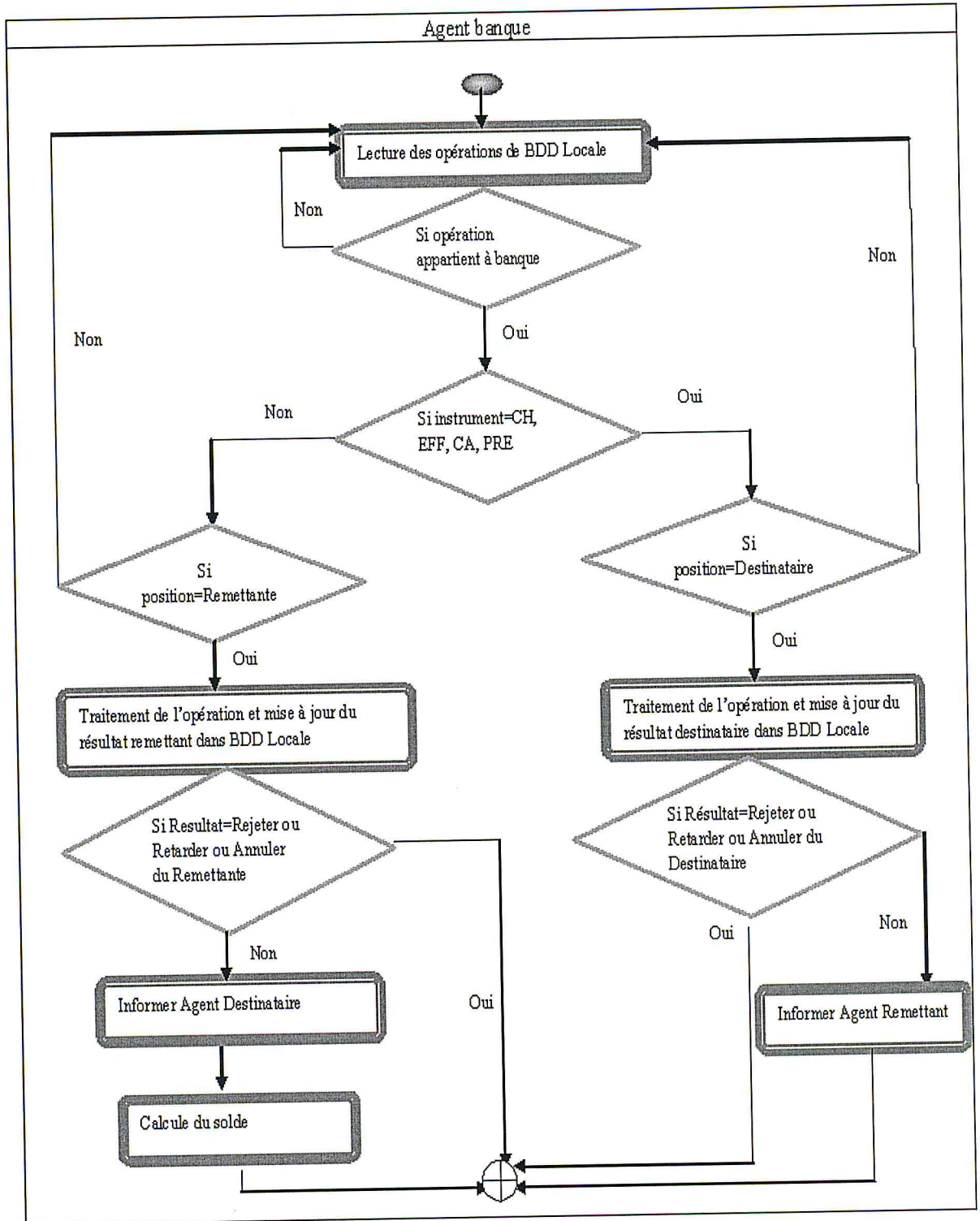


Fig. VI.11 Diagramme d'activité des traitements de l'agent banque (UML).

5.6. Les diagrammes d'interaction

5.6.1. Diagramme d'interaction « Ouverture de la journée de compensation »

Avant chaque début de journée de compensation, l'agent ATCI doit mettre au courant tous les agents banques de l'ouverture de la journée de compensation, pour s'initialiser. À noter que l'agent ATCI informe les agents banques du début de chaque séance de compensation, sauf la première séance, qui est égale au début de journée de compensation.

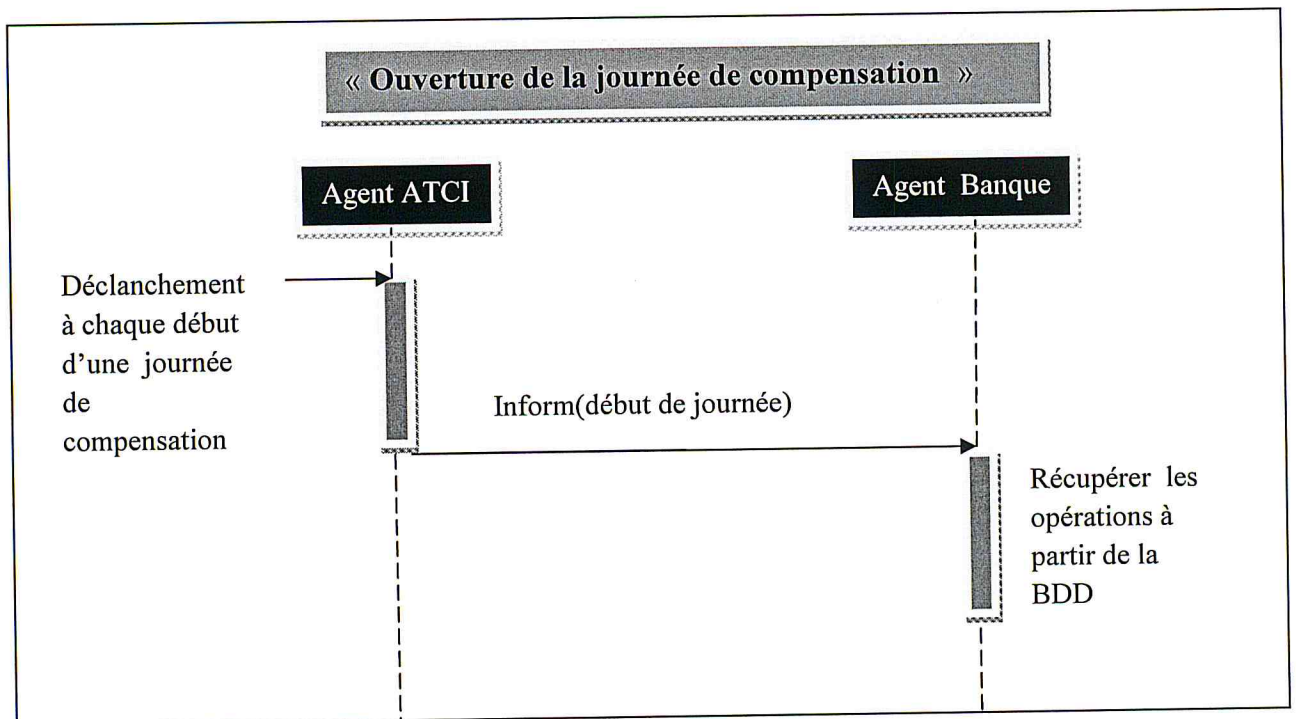


Fig. VI.12 Diagramme d'interaction « Ouverture de la journée de compensation ».

5.6.2. Diagramme d'interaction « Fermeture de la journée de compensation »

Avant chaque fin de journée de compensation, l'agent ATCI doit prévenir tous les agents banques. À noter que l'agent ATCI informe les agents banque de la fin de chaque séance de compensation ; a fin de procéder a des mise a jour (décrémentation des dates de règlement, donner un statut à l'état précédent(Créditrice ou bien Débitrice) calculé en fonction du solde de la journée...).

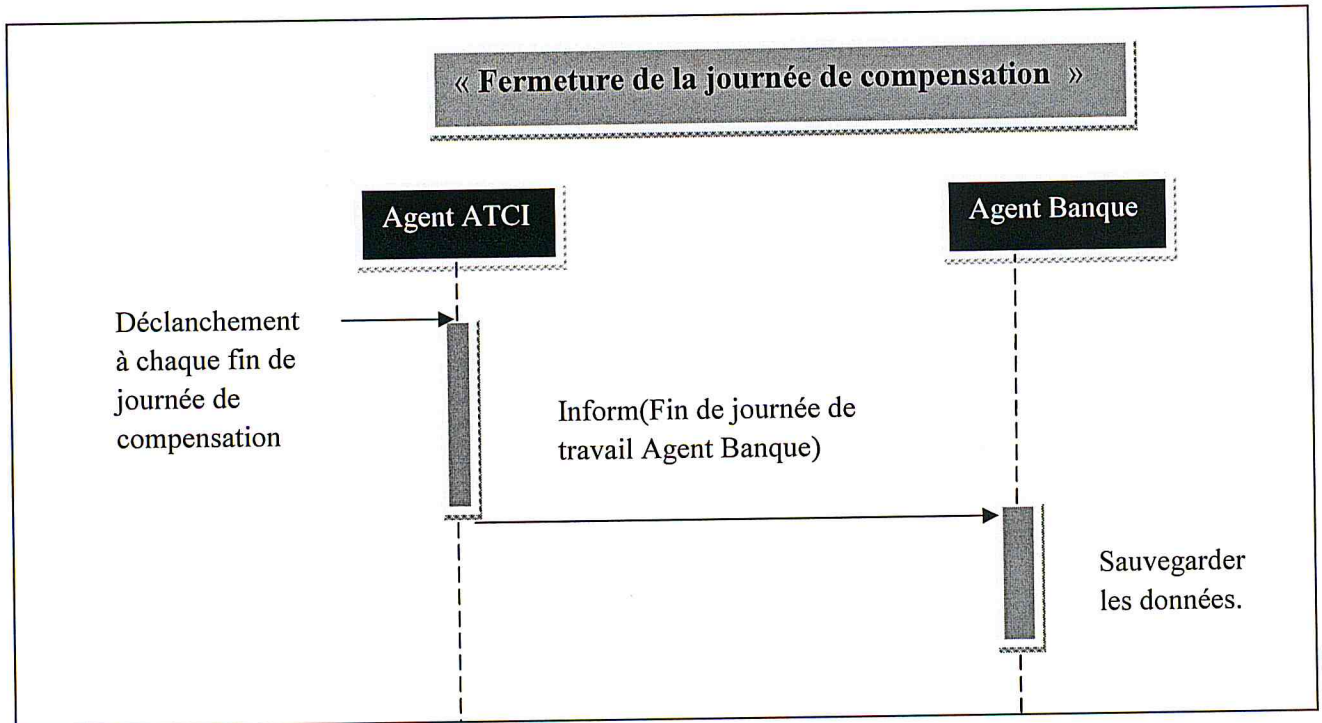


Fig. VI.13 Diagramme d'interaction « Fermeture de la journée de compensation ».

5.6.3. Diagramme d'interaction « informer AB Remettante »

Le processus « informer AB Remettante » est déclenché à chaque fois, qu'un Agent Banque Destinataire traite une opération qui aboutit à un état transitoire qui peut avoir une des quatre possibilités (retarder, accepter, rejeter, annuler), mais notre intérêt repose sur l'état accepter ou l'agent destinataire doit informer l'agent remettant pour la mise à jour des soldes.

Un autre cas de figure qui se présente est celui de confirmation et validation de l'état définitif (rejeter, accepter) de l'opération échangée, où on s'intéresse au cas où il y'a un rejet et cela pour que l'agent banque remettant mis à jour le solde.

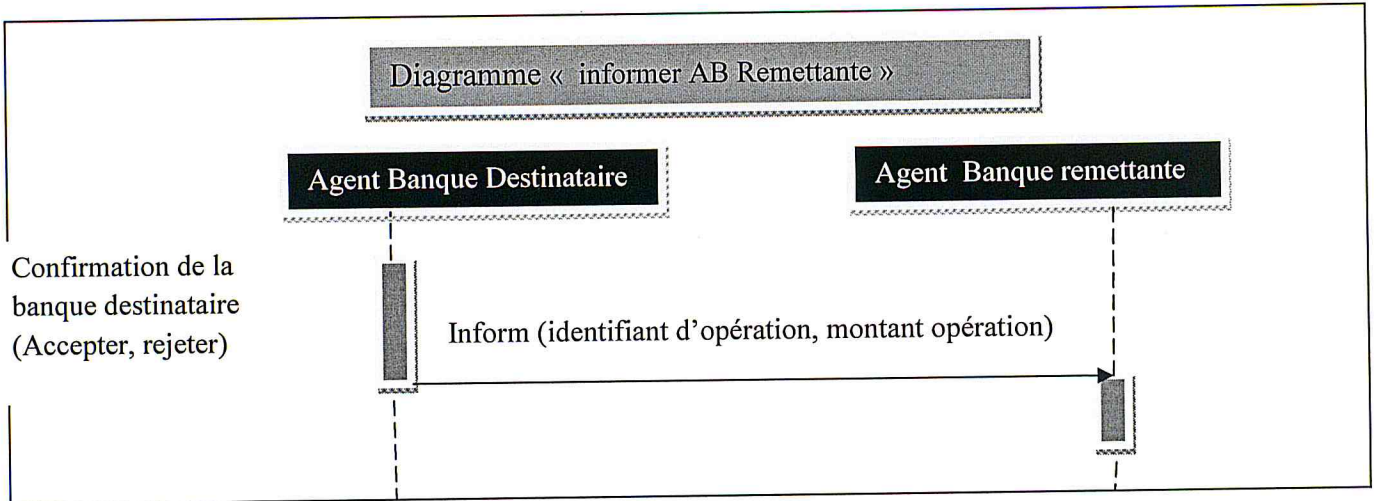


Fig.VI.14 Diagramme d'interaction « informer AB Remettante ».

5.6.4. Diagramme d'interaction « informer AB Destinataire »

Le processus « informer AB destinataire » est déclenché à chaque fois, qu'un Agent Banque Remettant traite une opération qui aboutit à un état transitoire qui peut avoir une des quatre possibilités (retarder, accepter, rejeter, annuler), mais notre intérêt repose sur l'état accepter ou l'agent remettant doit informer l'agent destinataire pour la mise à jour des soldes.

Un autre cas de figure qui se présente est celui de confirmation et validation de l'état définitif (rejeter, accepter) de l'opération échangée, où on s'intéresse au cas où il y'a un rejet et cela pour que l'agent banque destinataire mis à jour le solde.

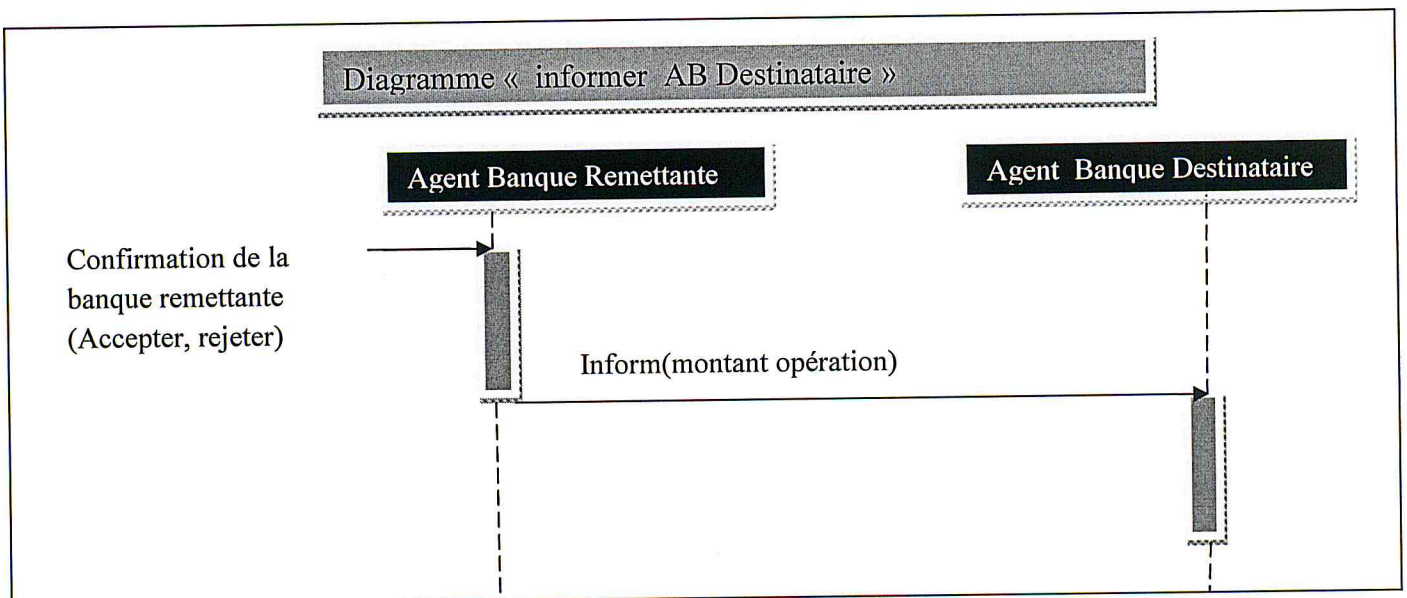


Fig.VI.15 Diagramme d'interaction « informer AB Destinataire ».

5.7. Conception de la Base de Données

5.7.1. Identification des classes de la BDD

Les classes pertinentes représentant les entités réelles et/ou abstraites constituant le domaine de notre étude sont les suivantes :

- **Classe « Banque »** : cette classe contient les informations relatives aux agents banques.
- **Classe « Opération »** : cette classe contient toutes les opérations échangées.
- **Classe « Type instrument »** : cette classe contient tous les types de moyenne de paiements interbancaires qui existent.

5.7.2. Diagramme de classe de la BDD

Notre diagramme de classe comporte 3 classes.

Classe Banque qui a deux liens différents avec la classe Opération car d'une part elle est remettante de l'opération et d'une autre part elle destinataire de l'opération.

La classe opération a un lien avec la classe Type instrument, car chaque opération possède un type d'instrument.

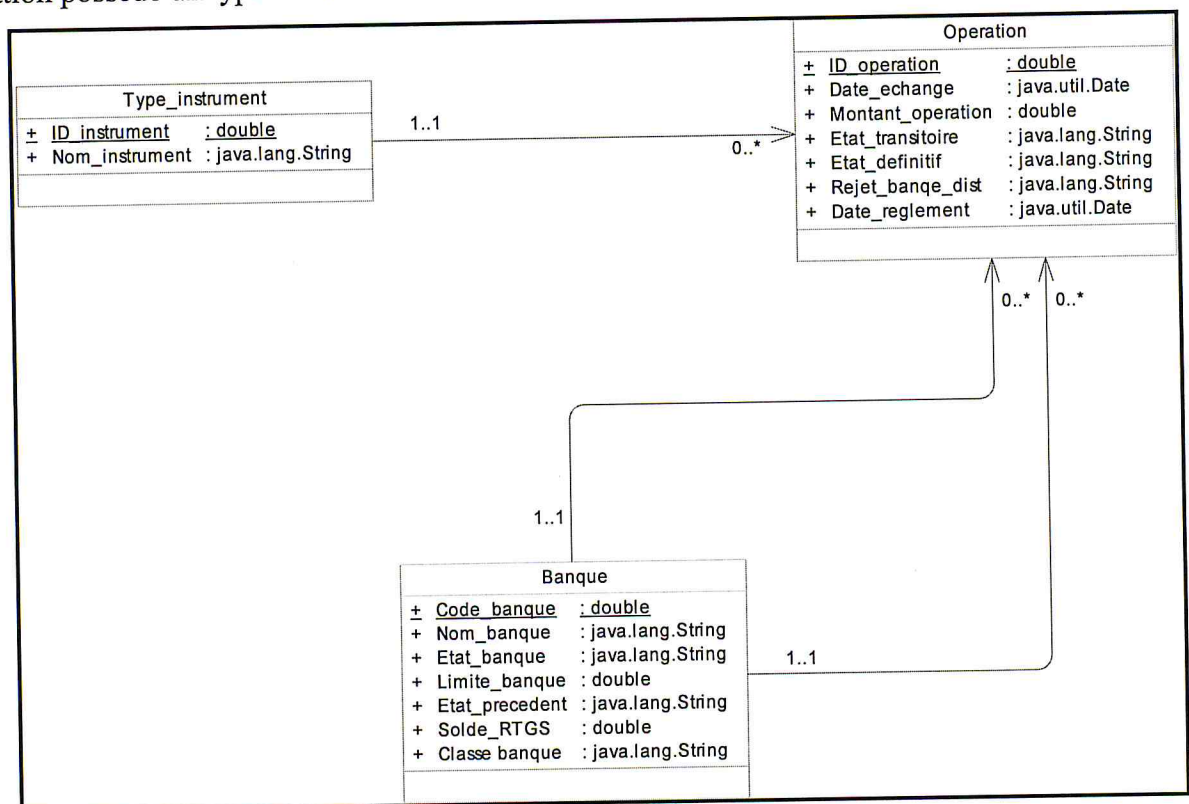


Fig. VI.16 Diagramme de classes de la base de données Locale (UML).

5.7.3. Description des classes

Classes	Attribut	Description
Banque	ID_Banque	Identifiant de l'Agent Banque
	Nom_Banque	Nom de l'Agent Banque
	Etat_Banque	Etat de l'agent banque
	Limite_Banque	Limite bancaire de l'Agent Banque
	Etat_precedent	Etat précédant de l'Agent Banque
	Solde_en_RTGS	Solde de l'Agent Banque au niveau du RTGS
	Classe_banque	La classe ou appartient l'Agent Banque
Opération	ID_operation	Identifiant de l'opération
	Date_echange	La date de l'opération
	Date_reglement	La date ou l'opération doit être régler
	Montant_operation	Montant de l'opération
	Etat_transitoir	Etat de l'opération en entré
	Etat_definitif	Etat définitif de l'opération
	Rejet_Banque_Dist	Le rejet bancaire de la banque destinataire.
type instrument	ID_instrument	Identifiant de l'instrument
	Nom_instrument	Nom de l'instrument

Tab VI.20 Description des classes de la base de données.

6. Conclusion

Dans ce chapitre nous avons présenté une conception détaillée de notre système en utilisant le langage UML, cette étude conceptuelle nous a permet de mettre en évidence les étapes nécessaires pour la création d'un système de suivi des échanges interbancaire.

La prochaine étape consiste donc en la concrétisation du ce que nous avons proposé.

1. Introduction

Après l'état de l'art autour des éléments nécessaires à la modélisation multi-agents d'un système ATCI, et après avoir conçu un modèle qui tente de répondre aux objectifs que nous nous sommes fixés, nous présentons dans cette partie l'implémentation de notre prototype. Pour ce faire, nous présenterons d'abord les outils utilisés : le langage de programmation, le SGBD et la plateforme multi-agents.

2. Environnement de développement

2.1. Plateforme Multi-agent

La plateforme utilisée pour le développement du système multi-agents est Jade. Jade est une plateforme de développement d'agents Open Source développée par CSELT² et qui résulte principalement d'une activité de recherche. Ces principales caractéristiques sont :

- JADE n'est couplée à aucune méthodologie de développement ;
- Les agents développés sous la plate-forme JADE, sont entièrement écrits en Java. Le choix de la plate forme s'est donc imposé comme étant une conséquence de notre choix en termes de langage de programmation Java.
- JADE simplifie l'implémentation d'un SMA à travers un MiddleWare³ répondant aux spécifications de la FIPA, une librairie de classes que les utilisateurs peuvent utiliser et étendre, ainsi qu'un ensemble d'outils graphiques qui permettent le débogage et l'administration du SMA.
- JADE assure une communication transparente par échange de messages dans le langage normalisé FIPA-ACL.
- JADE diminue l'effort de programmation car elle implémente des agents (le DF « Directory Facilitator », le AMS « Agent Management System » et le Sniffer) dont les fonctionnalités sont utiles à notre application. Ces agents représentent respectivement les services de pages jaunes, de pages blanches et du service d'écoute.
- JADE gère le cycle de vie des agents dès leur naissance jusqu'à leur destruction (création, mort, reprise, ...).

² Centro Studi E Laboratori Telecomunicazioni, traduit par Centre d'Etude et Laboratoire de Télécommunication : <http://jade.cselt.it>

³ Logiciel Médiateur : Mécanisme assurant le dialogue entre différentes applications ou portions d'une même application réparties sur plusieurs postes, clients ou serveurs.

2.2.Langage de programmation

Afin de réaliser notre SMA et ses interfaces, nous avons choisi le langage JAVA, ce choix se justifie par :

- Les agents développés sous JADE sont entièrement écrits en JAVA. Ce langage s'est donc imposé comme étant une conséquence du choix précédent.
- JAVA est un langage multiplateforme qui permet aux concepteurs d'écrire un code capable de fonctionner dans tous les environnements (quelque soit le système d'exploitation).
- JAVA est un langage orienté objet, qui simplifie l'écriture du code.
- JAVA est doté d'une riche bibliothèque de classes comprenant la gestion des interfaces graphiques (fenêtre, boîte de dialogue).
- Un accès simplifié aux bases de données, soit à travers la passerelle JDBC-ODBC ou à travers un pilote JDBC spécifique au SGBD.

2.3.SGBD :

Notre système possède une base de données qui contient les données du système ATCI telles que : les opérations échanger entre les banques, la limite financières de chaque banque, etc. Vu la quantité énorme et l'importance des données stockées dans cette base de données, nous avons opté pour le leader mondial de base de données **ORACLE** pour sa robustesse et sa sécurité, nous utiliserons la version 9g.

2.4.Module d'apprentissage

Nos agents banques comportent tous un module d'apprentissage et de raisonnement basé sur les systèmes de classeurs qui leur permet de prendre des décisions et d'apprendre de leurs expériences (apprentissage par renforcement). Un tel module est souvent très complexe à implémenter c'est pour cela que notre choix s'est porté sur le package **ART**⁴ (pour **Artificial Reasoning Toolkit**) qui est une bibliothèque de classes Java consacrée à la manipulation des Algorithmes Génétiques et des Systèmes de Classeurs.

ART facilite l'obtention d'agents intelligents entièrement autonomes, capables de gérer leurs propres comportements et de les adapter dans les

⁴ ART : <http://eco83.econ.unito.it/golem/>

différentes situations environnementales. Il permet de paramétrer les systèmes de classeurs grâce à la classe **CsManager** qui est la classe principale permettant d'effectuer toutes les opérations de base pour un usage standard d'un système de classeurs .

3. Les algorithmes d'exécutions

3.1. Algorithme d'exécution de l'Agent ATCI

Début

Pour chaque journée de compensation

// initialisation des paramètres (nombre de banques, etc.)

Pour chaque journée de compensation //à travers un cyclicBehaviour

Faire

si (heure=13 :30) *alors*

M := début de nouvelle journée de compensation;

Envoyer message M ;

sinon

si (heure=17 :30) *alors*

M :=fin de la première séance de compensation ;

Envoyer message M aux Agents Banque ;

Génération du fichier OUTGO ;

sinon

si (heure=17 :31) *alors*

M :=début de la deuxième séance de compensation ;

Envoyer message M aux Agents Banque ;

sinon

si (heure=7 :30) *alors*

M :=fin de la deuxième séance de compensation ;

Envoyer message M aux Agents Banque ;

Génération du fichier OUTGO ;

sinon

si (heure=7 :31) *alors*

M :=début de la troisième séance de compensation ;

Envoyer message M aux Agents Banque ;

```

    sinon
        si(heure=11 :00) alors
            Mise à jour la base de données locale ;
            M :=fin de la troisième séance de compensation ;
            Envoyer message M aux Agents Banque ;
            Génération du fichier OUTGO ;
        sinon
            si(arrivée opération) alors
                M :=arrivée opération ;
                Envoyer message M aux Agents Banque ;
            fsi
        fsi
    fsi
    fsi
    fsi
    fait
```

3.2.FinAlgorithme d'exécution de l'agent banque

Début

//initiation des paramètres (nombre de banque, etc....)

Pour chaque journée de compensation

Faire

M := recevoir message//à travers un cyclicBehaviour

si M contient «début de journée de compensation » alors

Appelle du behaviour début de journée de compensation ;

sinon

si M contient «début de la deuxième séance de compensation » alors

Appelle du behaviour début de deuxième séance de compensation ;

sinon

si M contient «début de la troisième séance de compensation» alors

Appelle du behaviour début troisième séance de compensation ;

sinon

si M contient « fin de séance de compensation » alors

Appelle du behaviour fin de séance de compensation ;

sinon

si M contient « fin de journée de compensation » alors

Appelle du behaviour fin de journée de compensation ;

sinon

si M contient « arrivée des opérations» alors

Appelle du behaviour arrivée opération ;

fsi

fsi

fsi

fsi

fsi

fsi

fait

Fin

Description du behaviour début de journée de compensation

Type du behaviour :oneShotBehaviour

Début

Tant que (Existe opération)

Faire

Récupération des champs nécessaire aux traitements ;

//adaptation des instruments

si (id Instrument <= 2) *alors*

Type instrument := cheque ;

sinon

si (id Instrument = 3) *alors*

Type instrument := prélèvement ;

sinon

si ((id Instrument = 4) ou (id Instrument = 5)) *alors*

Type instrument := carte ;

sinon

si ((id Instrument = 6) ou (id Instrument = 7)) *alors*

Type instrument := effet ;

sinon

si (id Instrument >= 8) *alors*

Type instrument := virement ;

fsi

fsi

fsi

fsi

fsi

//attribution des valeurs aux Solde1, Solde2, Solde3

si (état définitif = accepter) *alors*

si (id Instrument < 8) *alors*

si (type banque = destinataire) *alors*

si (date règlement = J+1) *alors*

Solde2 := Solde2 - montant opération ;

sinon

Solde1 := Solde1 - montant opération ;

fsi

sinon

si (date règlement = J+1) *alors*

Solde2 := Solde2+ montant opération ;

sinon

Solde1 := Solde1+ montant opération ;

fsi

fsi

sinon

si (type banque = destinataire) *alors*

Solde1 := Solde1+ montant opération ;

sinon

Solde1 := Solde1- montant opération ;

fsi

fsi

sinon

si (id Instrument < 8) *alors*

si (état transitoire = accepter) *alors*

si (type banque = destinataire) *alors*

si (date règlement = J) *alors*

Solde1 := Solde1 - montant opération ;

sinon

si (date règlement = J+1) *alors*

Solde2 := Solde2- montant opération ;

sinon

Solde3 := Solde3-montant opération ;

fsi

fsi

sinon

si (date règlement = J) *alors*

Solde1 := Solde1+ montant opération ;

sinon

si (date règlement = J+1) *alors*

Solde2 := Solde2 + montant opération ;

sinon

Solde3 := Solde3 + montant opération ;

fsi

fsi

fsi

sinon

si (type banque = destinataire) *alors*

si (date règlement = J) *alors*

Solde1 := Solde1 - montant opération ;

sinon

si (date règlement = J+1) *alors*

Solde2 := Solde2 - montant opération ;

sinon

Solde3 := Solde3 - montant opération ;

fsi

fsi

fsi

fsi

sinon

si (type banque = remettante) alors

Solde1 := Solde1 - montant opération ;

fsi

fsi

fsi

si ((type banque = destinataire) et (id instrument < 8) ou (type banque = remettante)

et (id Instrument >= 8)) alors

si(état transitoire=rejeter)ou état transitoire=retarder)ou état transitoire = annuler) alors

Préparation des paramètres CS ;

Traitement et génération d'un cycle CS ;

Calcul du reward ;

Attribution du reward ;

si (décision = accepter) alors

si ((type banque = destinataire) et (id Instrument < 8)) alors

Informer agent banque remettante ;

sinon

Informer agent banque destinataire ;

fsi

fsi

Mise à jour de l'état transitoire de l'opération ;

fsi

fsi

Fait

Fin

Description du comportement Arrivée des opérations

Type du comportement :cyclicBehaviour

Début

Récupération des champs nécessaire aux traitements ;

//adaptation des instruments

si (id Instrument <= 2) *alors*

 Type instrument := cheque ;

sinon

si (id Instrument = 3) *alors*

 Type instrument := prélèvement ;

sinon

si ((id Instrument = 4) ou (id Instrument = 5)) *alors*

 Type instrument := carte ;

sinon

si ((id Instrument = 6) ou (id Instrument = 7)) *alors*

 Type instrument := effet ;

sinon

si (id Instrument >= 8) *alors*

 Type instrument := virement ;

fsi

fsi

fsi

fsi

fsi

//param est l'indice qui définit la tâche à faire : mise à jour d'état ou bien une nouvelle opération insérée

//tel que si param=0 implique qu'il s'agit d'une opération de mise à jour

//si non il s'agit de traiter une nouvelle opération

si (param = 0) *alors*

si (id Instrument < 8) *alors*

si (type banque = destinataire) *alors*

si (rejet banque dis = oui) *alors*

Insertion de l'opération dans la table HISTORIQUE ;

Suppression de l'opération depuis la table opération ;

si (date de règlement = J) *alors*

Solde1 := Solde1 + montant opération ;

Informez l'autre agent banque pour le retrait du Solde1 ;

Sinon

si (date de règlement = J+1) *alors*

Solde2 := Solde2 + montant opération ;

Informez l'autre agent banque pour le retrait du

Solde2 ;

sinon

Solde3 := Solde3 + montant opération ;

Informez l'autre agent banque pour le retrait du

Solde3 ;

fsi

fsi

sinon

si (rejet banque dis = non) *alors*

Mise à jour de l'état définitif de l'opération ;

Mise à jour de la date de règlement de l'opération ;

sinon

si (date règlement = J) *alors*

Solde1 := Solde1 - montant opération ;

sinon

si (date règlement = J+1) *alors*

Solde2 := Solde2 - montant opération ;

sinon

Solde3 := Solde3 - montant opération ;

fsi

fsi

fsi

fsi

fsi

Sinon

si (type banque = remettante) *alors*

si (rejet banque dis = oui) *alors*

Insertion de l'opération dans la table HISTORIQUE ;

Suppression de l'opération depuis la table OPERATION ;

Solde1 := Solde1 + montant opération ;

Informez l'autre agent banque concerné pour la mise à jour du Solde1 ;

Sinon

si (rejet banque dis = non) *alors*

Mise à jour de l'état définitif dans la table OPERATION ;

sinon

Solde1 := Solde1 - montant opération ;

fsi

fsi

fsi

fsi

sinon

//récupération des opérations récemment arrivée une par une

si (état transitoire = nul) *alors*

Préparation des paramètres CS ;

Génération du Cycle CS ;

si (décision = accepter) *alors*

si ((type banque = destinataire) et (id Instrument<8)) *alors*

Informer agent banque remettante ;

sinon

Informer agent banque destinataire ;

fsi

fsi

Mise à jour de l'état transitoire ;

fsi

Fin

4. Architecture de l'application :

Notre application est divisée en deux grandes parties :

- Une partie qui assure toutes les fonctionnalités nécessaire.
- Une partie supplémentaire qui offre des services de suivi et de mise à jour.

Dans ce qui suit nous verrons plus en détaille les composants de notre application.

5. Présentation de l'application

5.1. Authentification

Cette interface représente la phase Authentification. Son rôle est de contrôler les accès aux services de l'application. Le bénéficiaire doit saisir un mot de passe et indique son poste ; et comme résultat fournit :

- Acceptation : amène au Gestionnaire de tâches pour la manipulation.
- Accès refusé.

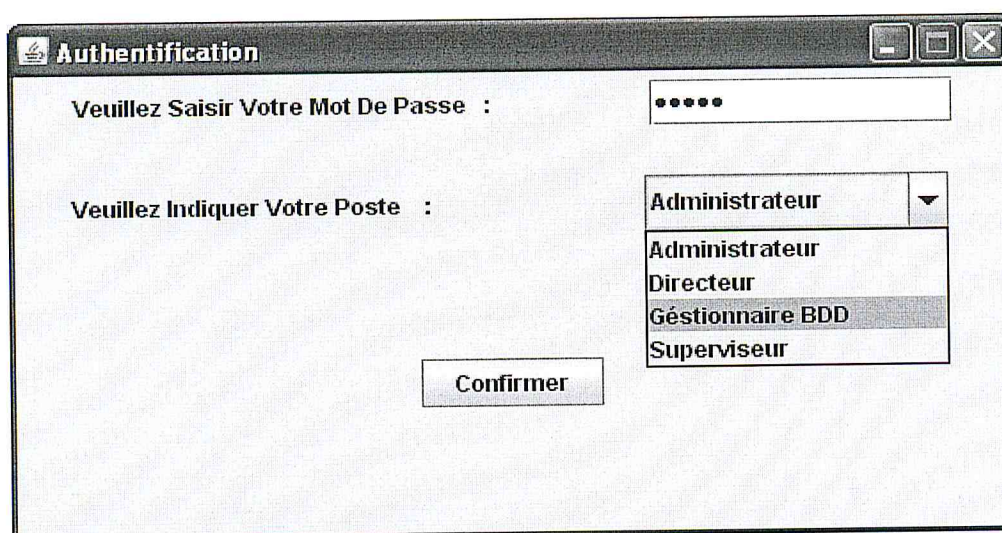


Fig.VII.1 Interface d'authentification.

5.2. Gestionnaire de tâches

Cette interface représente le cœur de notre application. Plusieurs tâches peuvent être réalisées grâce aux composants Swing suivants :

Numéro	Nature	Description
1	JButton	Génération des graphes d'évolution des soldes avec système actuel par journée de compensation.
2	JButton	Génération des graphes d'évolution des soldes avec notre système par journée de compensation.
3	JButton	Génération des graphes d'évolution des soldes de compensation avec système actuel

		pour plusieurs journées de compensation.
4	JButton	Génération des graphes d'évolution des soldes de compensation avec notre système pour plusieurs journées de compensation.
5	JButton	Génération d'un OUTGO sous format XML qui englobe toutes les opérations traitées durant une séance de compensation.
6	JButton	Donne la possibilité de voir les changements affecter aux opérations en temps réel.
7	JRadioButton	Permet de sélectionner un choix parmi plusieurs proposées.
8	JComboBox	Permet de déclencher un des processus de compensation.
9	JTextField	Permet d'afficher les traitements élaborés par les agents ainsi que leurs communications.
10	JButton	Pour quitter l'application
11	JMenuItem	Choisir la table à afficher.

Tab VII.1 Description des composants de la fenêtre Gestionnaire de tâches

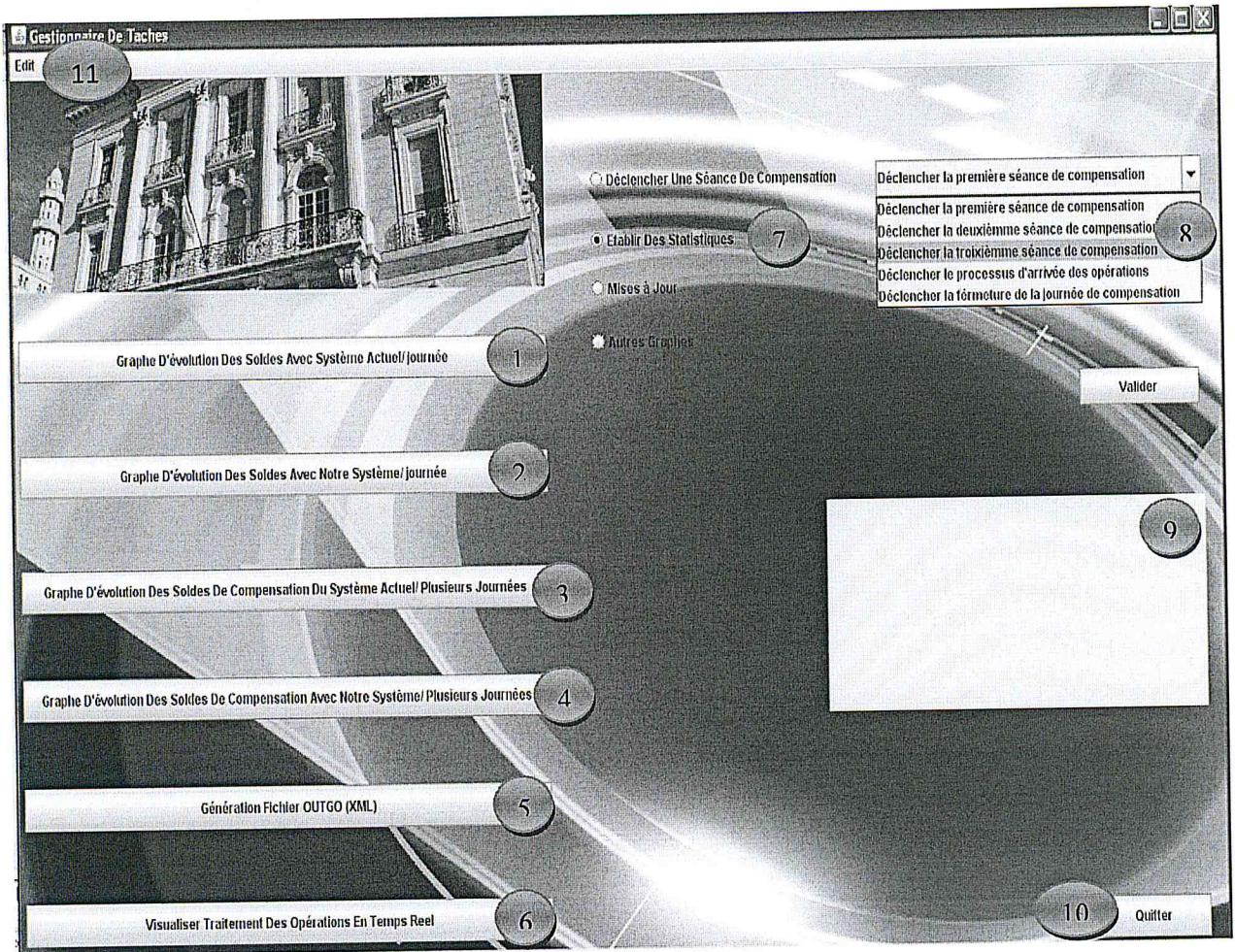


Fig.VII.2 Interface Gestionnaire de tâches

5.3. Statistiques

Permet de faire des statistiques sur plusieurs axes

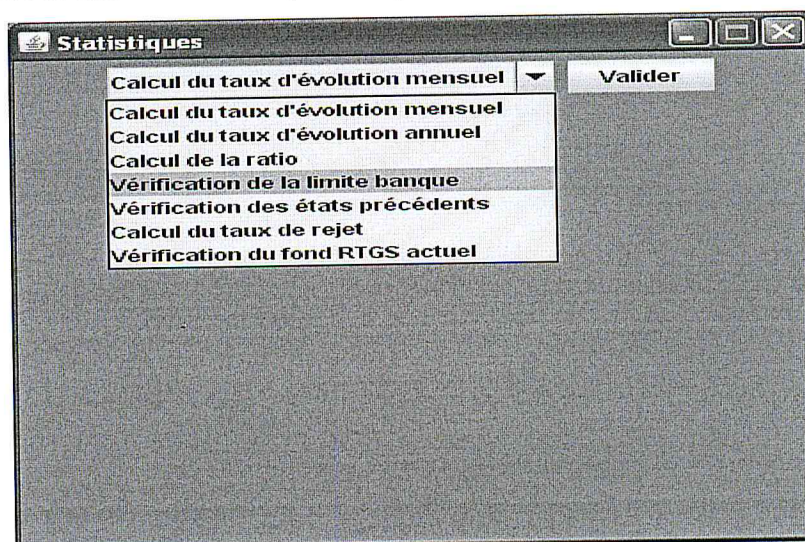


Fig.VII.3 Interface Statistiques

5.4.Taux d'Evolution Annuel

Permet de s'informer sur le Taux d'évolution annuel pour savoir l'importance de l'ensemble des opérations traitées entre deux années successives.

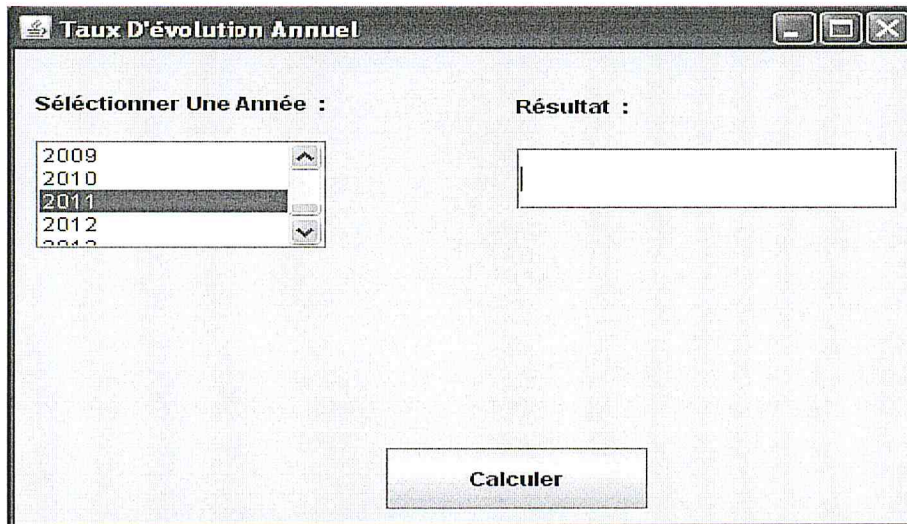


Fig.VII.4 Interface Taux d'Evolution Annuel

5.5.Taux d'Evolution Mensuel

Permet de s'informer sur le Taux d'évolution mensuel pour savoir l'importance de l'ensemble des opérations traitées entre deux mois successifs.

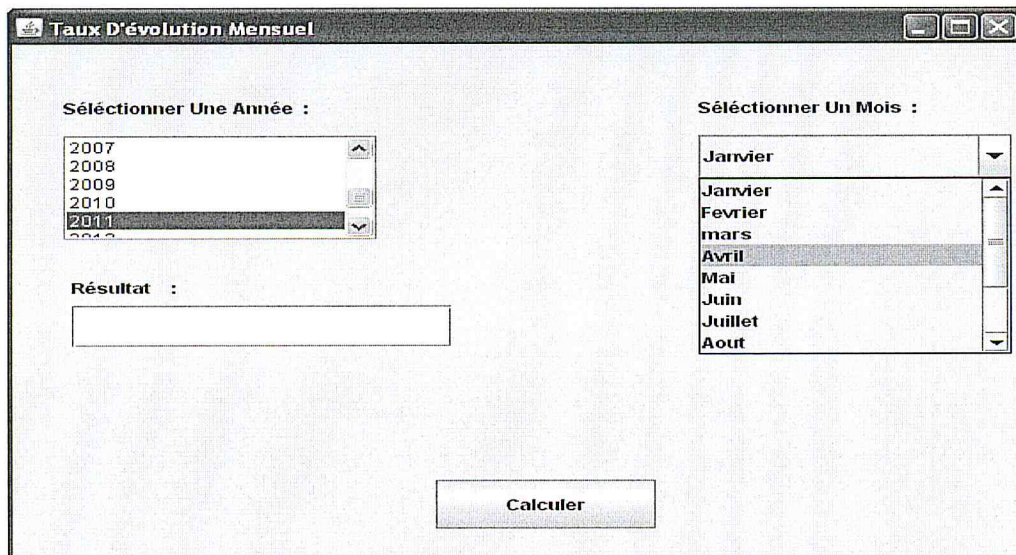


Fig.VII.5 Interface Taux d'Evolution Mensuel

5.6. Taux de Rejet

Permet de savoir le nombre des opérations rejetées pour la journée courante pour une banque sélectionnée ; ainsi que les rejets de banque à banque.

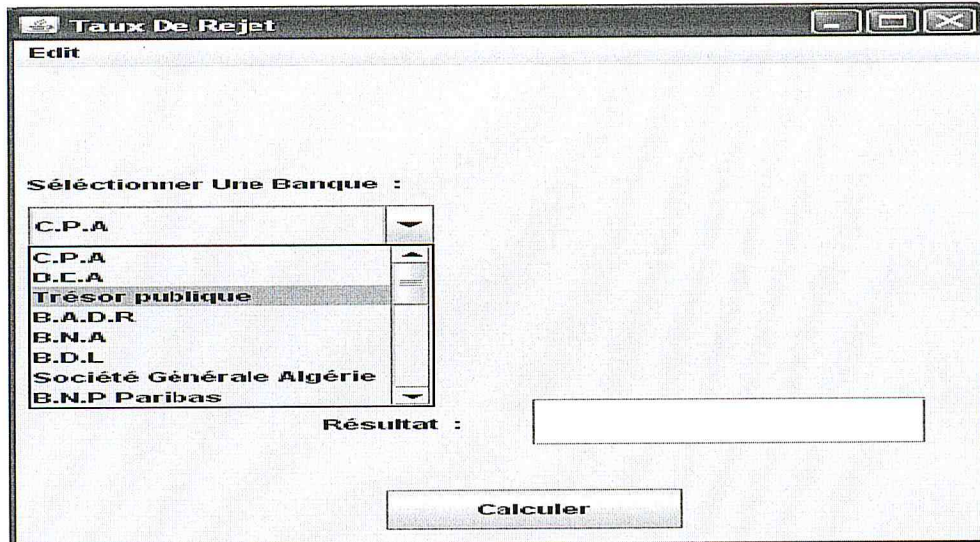


Fig.VII.6 Interface Taux de Rejet

5.7. Etat Précédent

Donne l'état de la journée précédente d'une banque choisi, en montant. Elle nous met en courant des états banque.

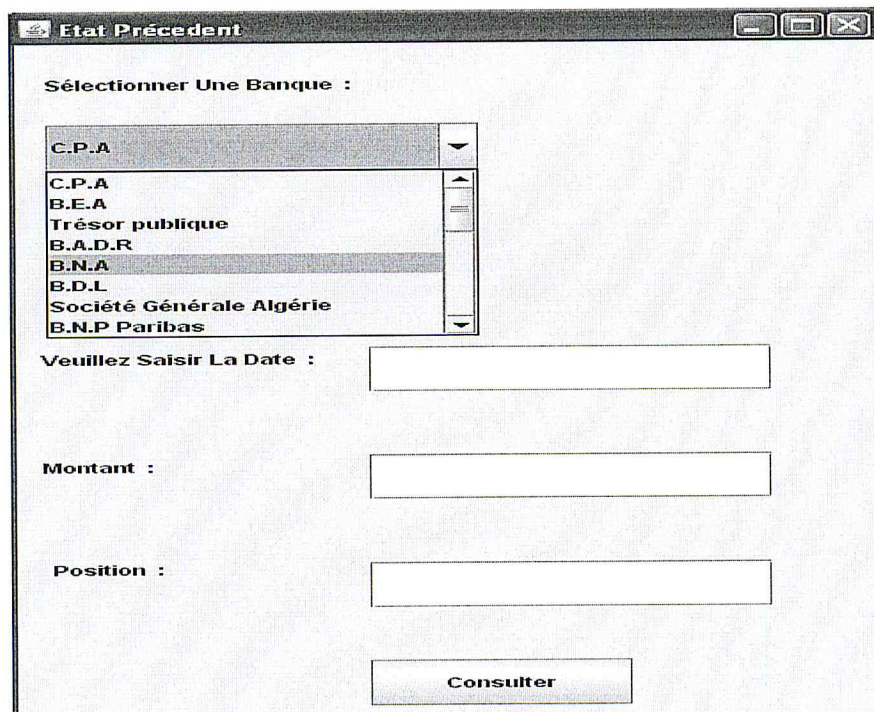


Fig.VII.7 Interface d'Etat Précédent

5.8. Limite Banque

Permet de connaître la limite banque d'une banque sélectionnée selon trois axes : solde1, solde2, solde3.

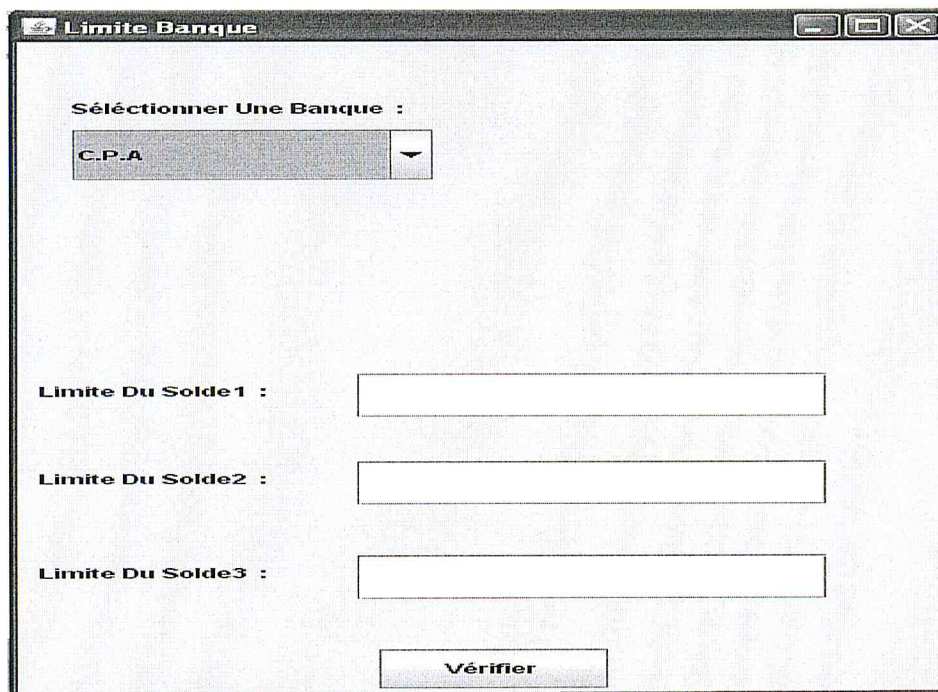


Fig.VII.8 Interface de la Limite Banque

5.9. La Ratio

Elle démontre l'instrument le plus convoité durant la journée de compensation.

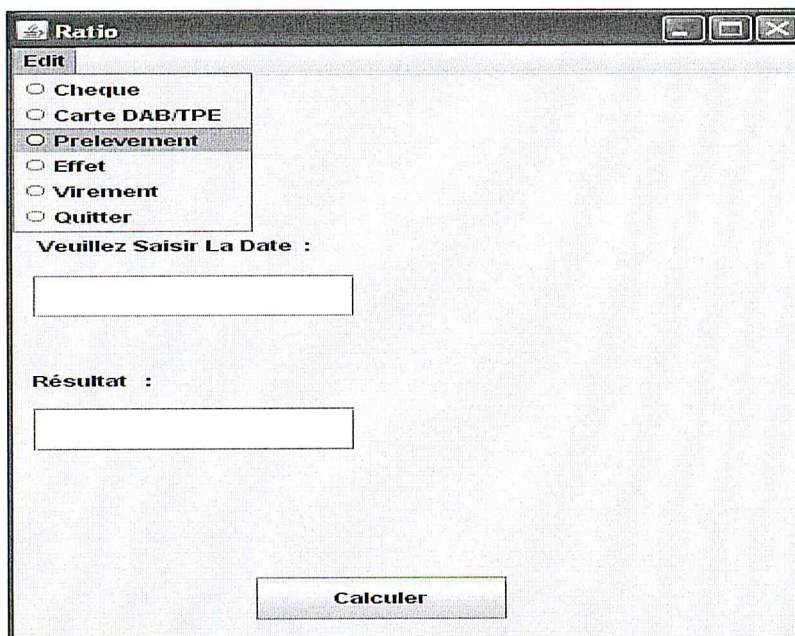


Fig.VII.9 Interface de calcul de la Ratio.

5.10. Contrôle

Elle gère les contrôles d'accès aux mises à jour de la base de données locale.

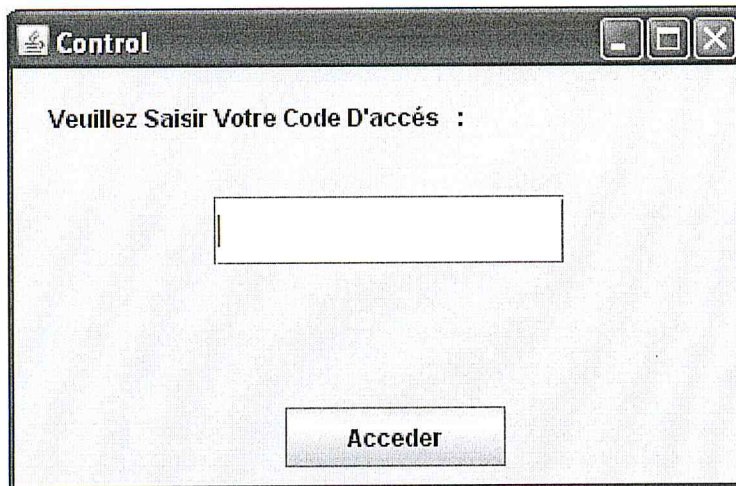


Fig.VII.10 Interface de Contrôle.

5.11. Catégorie Banque

Permet de sélectionner une tâche (ajout, suppression, modification...).

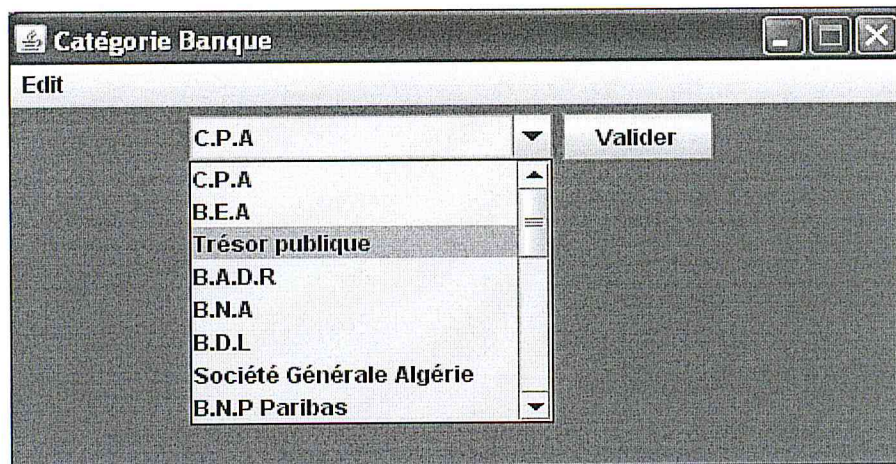


Fig.VII.11 Interface Catégorie Banque

5.12. Modification Catégorie banque

Permet de modifier la classe de la banque.



Fig.VII.12 Interface Modification Catégorie Banque.

5.13. Ajout

Permet d'ajouter des banques.

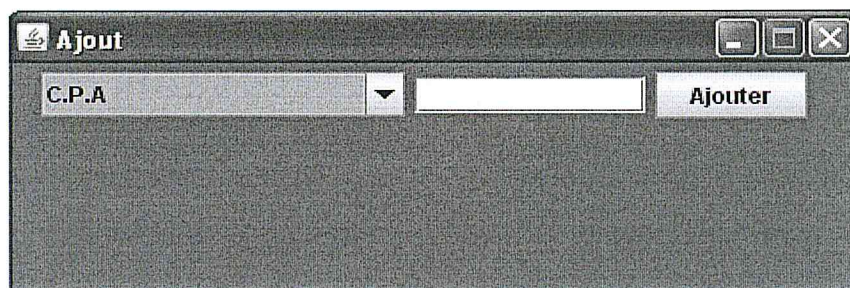


Fig.VII.13 Interface Ajout.

5.14. Modification Nom Banque

Permet de modifier le nom d'une banque.



Fig.VII.14 Interface Modification Nom Banque

5.15. Suppression

Permet de supprimer une banque.

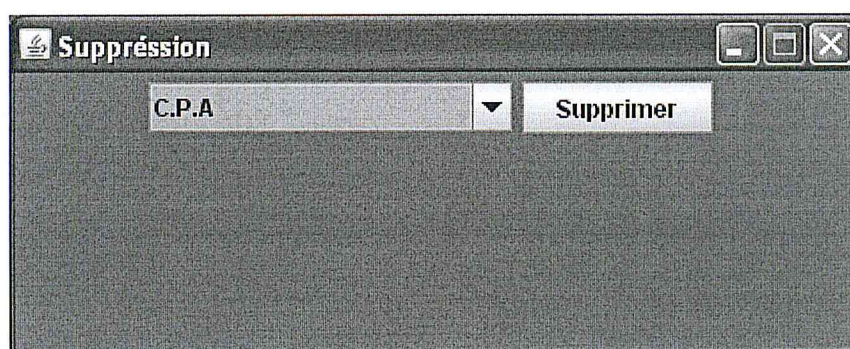


Fig.VII.15 Interface suppression d'une banque.

5.16. Génération des graphes

Donne la possibilité de générer des graphes selon plusieurs aspects.

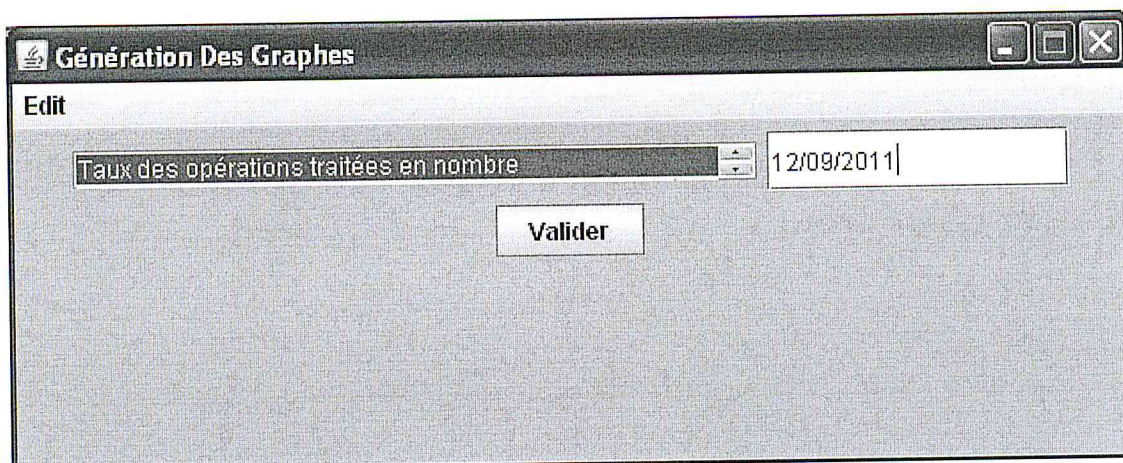


Fig.VII.16 Interface Génération des Graphes.

5.17. Taux des Opérations traitées en Nombre.

Ce graphe a pour but de montrer sous forme de barre les différents niveaux de traitement des opérations selon leur nature (chèque, virement, prélèvement, carte, effet) par journée.

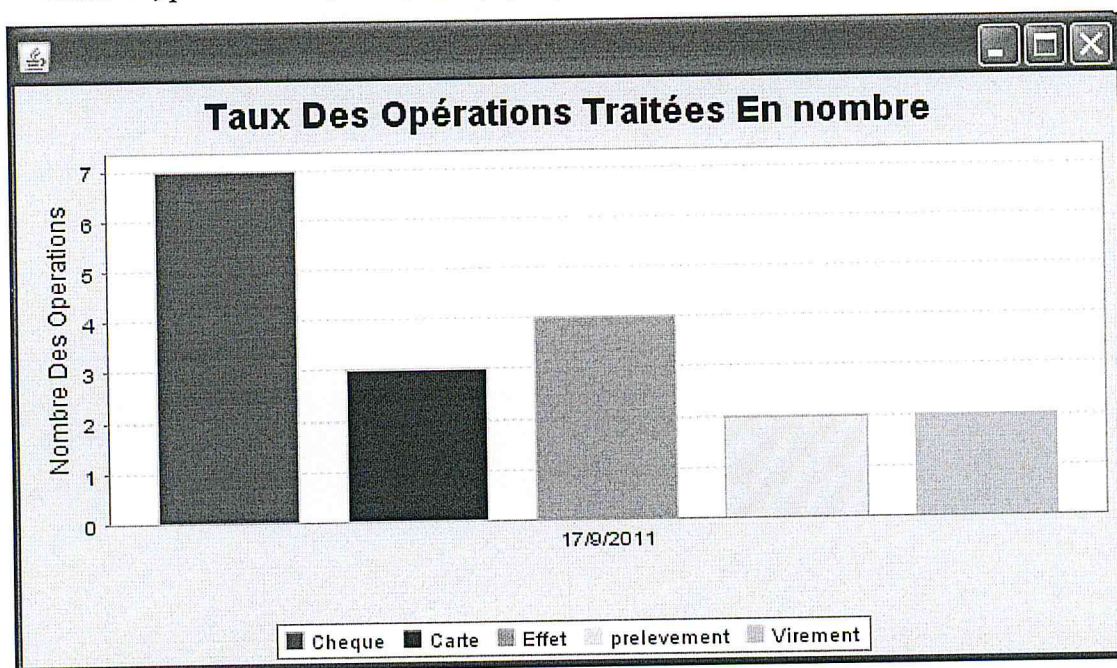


Fig.VII.17 Interface Taux des Opérations traitées en Nombre

5.18. Soldes virtuels des banques avec le système actuel

Ce type de graphe est généré après chaque fin de séance de compensation. Il représente les niveaux du solde virtuel atteint pour chaque banque et par journée de compensation. Les soldes virtuels sont calculés sans faire de traitements au préalable, c'est-à-dire toutes les opérations sont acceptées.

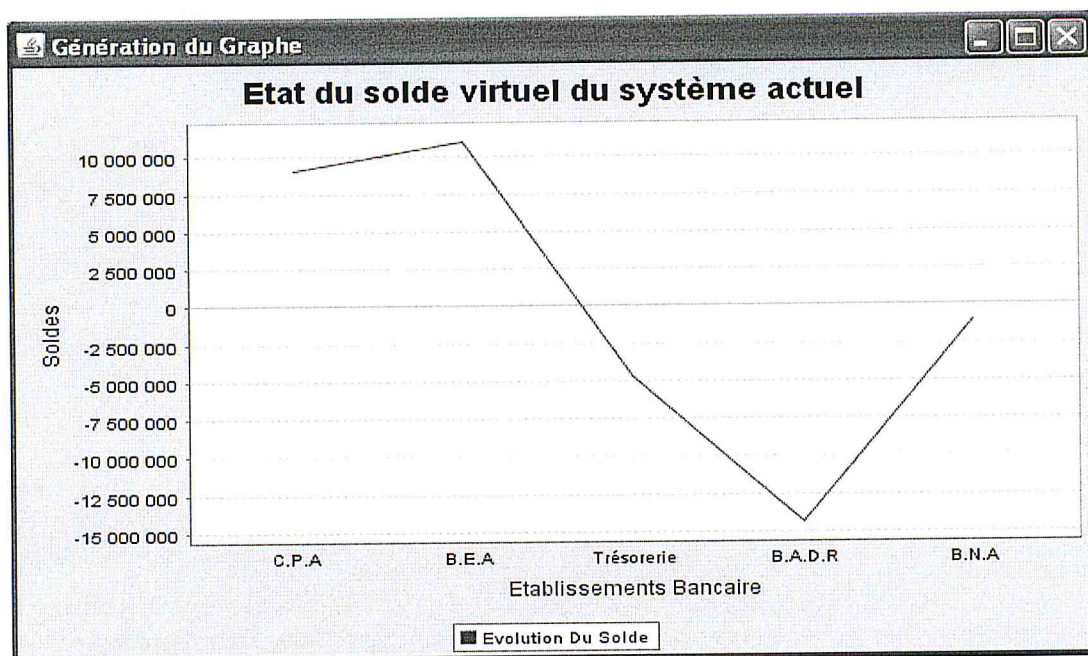


Fig.VII.18 Graphe des soldes virtuels par banque avec le système actuel/journée

5.19. Evolution du solde virtuel avec notre système

Ce type de graphe est généré Après chaque fin de séance de compensation. Il représente les niveaux du solde virtuel atteint pour chaque banque.les soldes sont générés par notre système de classeur.

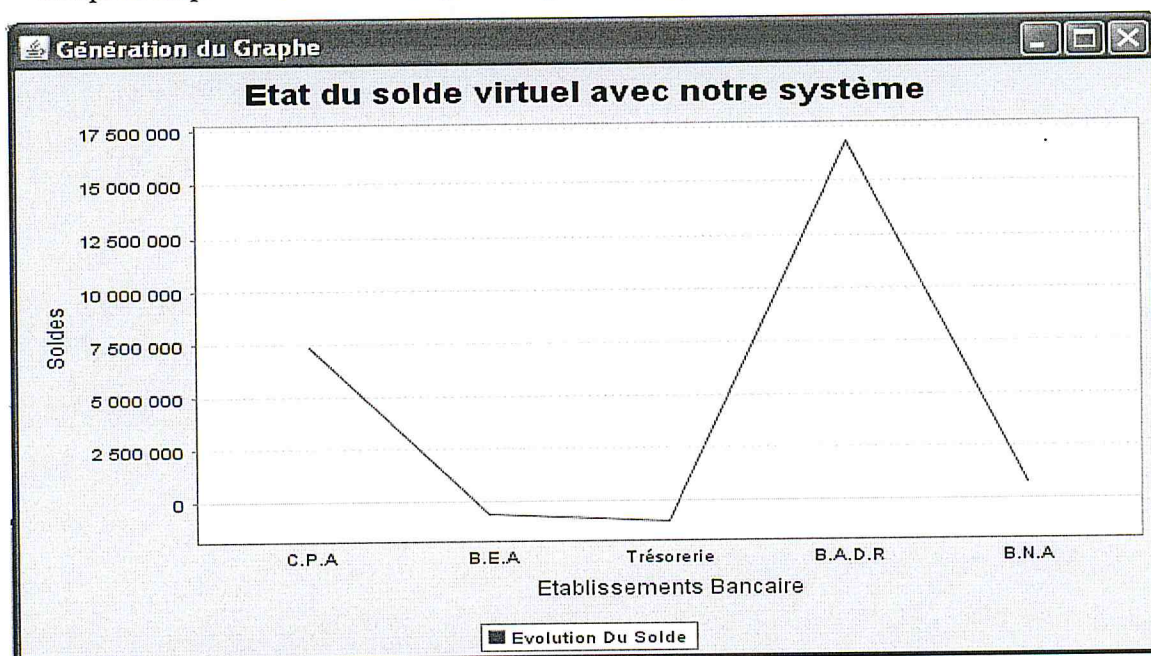


Fig.VII.19 Graphe des Soldes virtuels des banques avec notre système/journée

Le but de ces deux graphes est de comparer l'efficacité de l'ancien système et le notre afin de juger la performance de notre système.

5.20. Taux des opérations traitées en nombre

Ce graphe informe des taux (nombre d'opérations traitées) pour chaque type d'instrument de paiement (chèque, virement, prélèvement, carte, effet) par journée de paiement; afin d'avoir une idée des instruments les plus fréquentés.

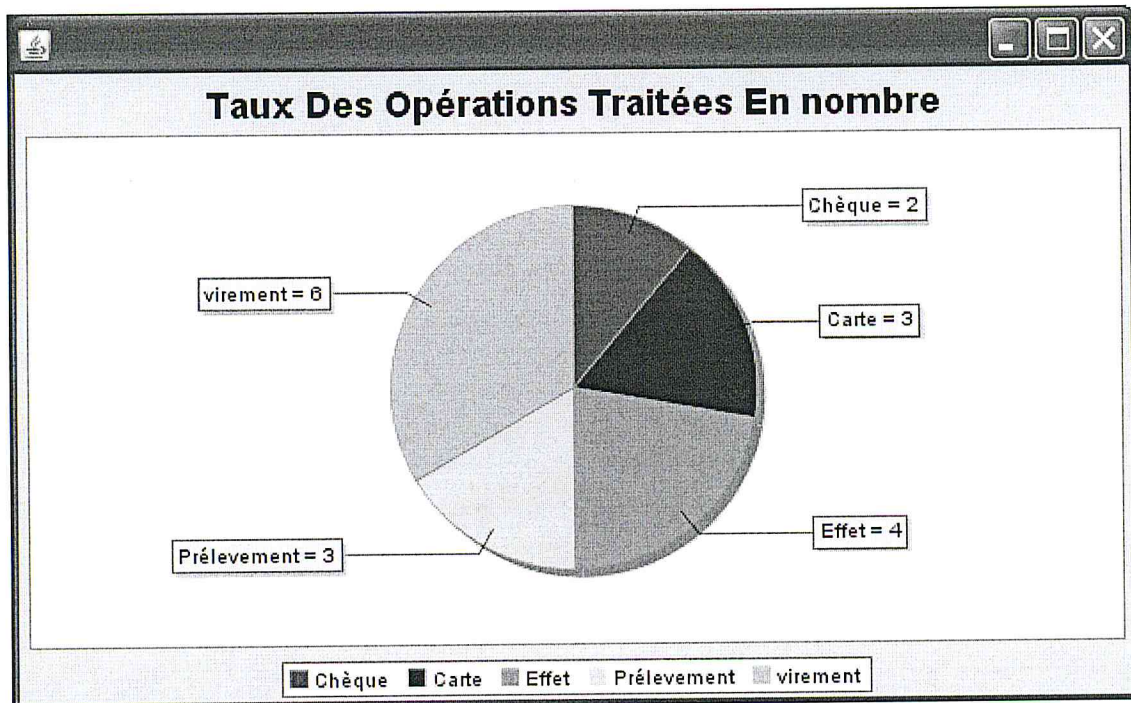


Fig.VII.20 Interface Taux des opérations traitées en nombre/journée

5.21. Evolution Annuelle du traitement des opérations

Ce graphe donne la possibilité aux responsables de visionner l'ensemble des opérations traitées durant les dix dernières années, afin de comparer l'état d'évolution des traitements des opérations de paiement pour chaque année. Ce graphe permet d'élaborer une classification des années, de l'année la plus élevée (concernant le traitement des opérations) à la plus faible.

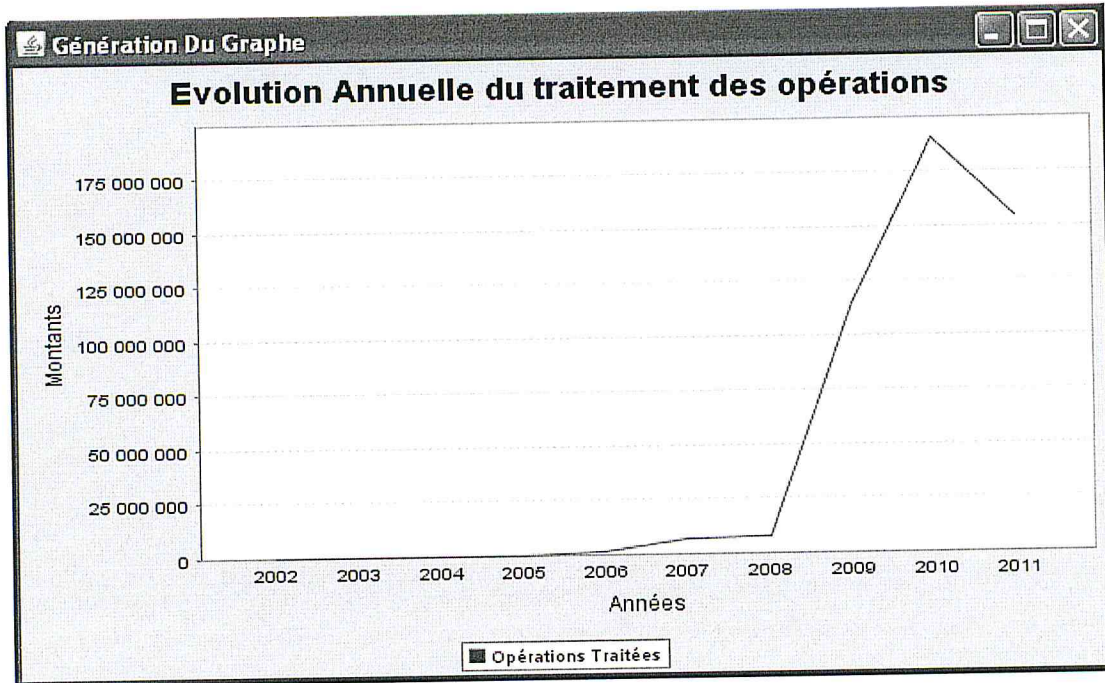


Fig.VII.21 Interface Evolution Annuelle des traitements des opérations

5.22. Evolution Mensuelle du traitement des opérations

Ce graphe donne la possibilité aux responsables de visionner l'ensemble des opérations traitées durant les mois de l'année, afin de comparer l'état d'évolution des traitements des opérations de paiement pour chaque mois. Ce graphe permet d'élaborer une classification des mois, du mois le plus élevé (concernant le traitement des opérations) au plus faible.

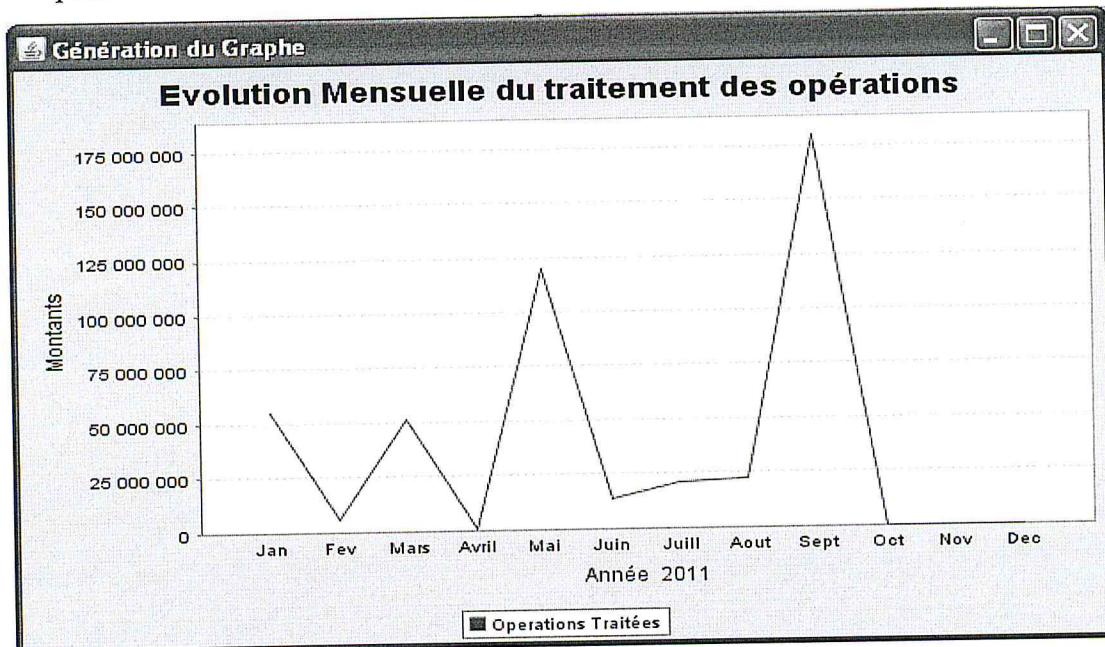


Fig.VII.22 Interface Evolution Mensuelle des traitements des opérations.

5.23. Evolution du solde virtuel Avec le Système Actuel

Le graphe suivant montre l'évolution des soldes virtuels des banques à travers une période de dix journées de compensation successifs. Ce graphe essaye de cibler les failles du système actuel en montrant l'évolution négative de certaines banques, qui peut causer des situations d'insolvabilité, si le problème persiste. Le système ne donne pas d'importance au nombre de banques qui présentent un solde négatif; et il n'attribue aucune alerte. Par conséquent, les soldes vont évoluer, même s'il s'agit d'un solde négatif, et ceci va certainement aggraver la situation financière des banques.

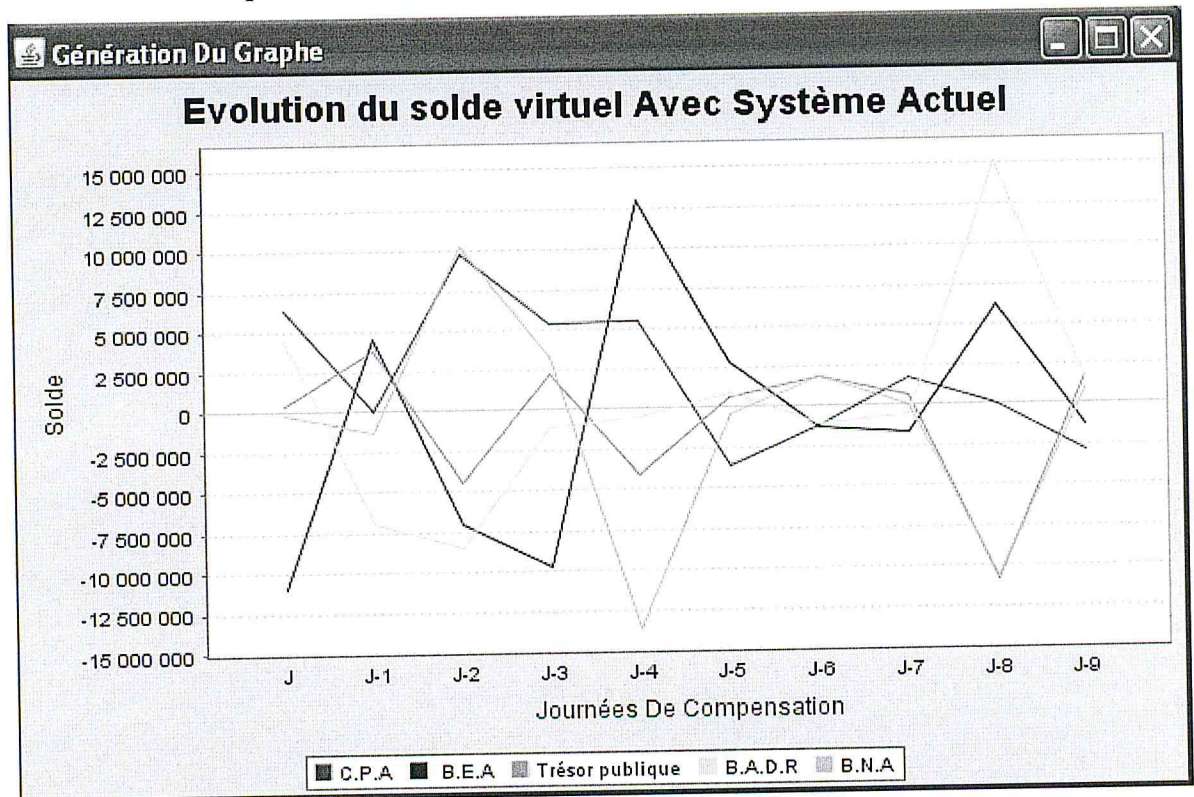


Fig. VII.23 Interface Evolution du solde virtuel Avec Système Actuel.

5.24. Evolution du solde virtuel Avec Notre Système

Ce graphe consiste à montrer l'évolution des soldes virtuels des banques à travers une période de dix journées de compensation. Ce graphe a pour objectif de montrer les améliorations apportées par notre système envers les traitements des soldes. Dans ce graphe, on remarque que notre système essaye d'équilibrer les soldes, et de réduire ainsi les marges entre les soldes négatifs et positifs pour maintenir la stabilité du système de compensation.

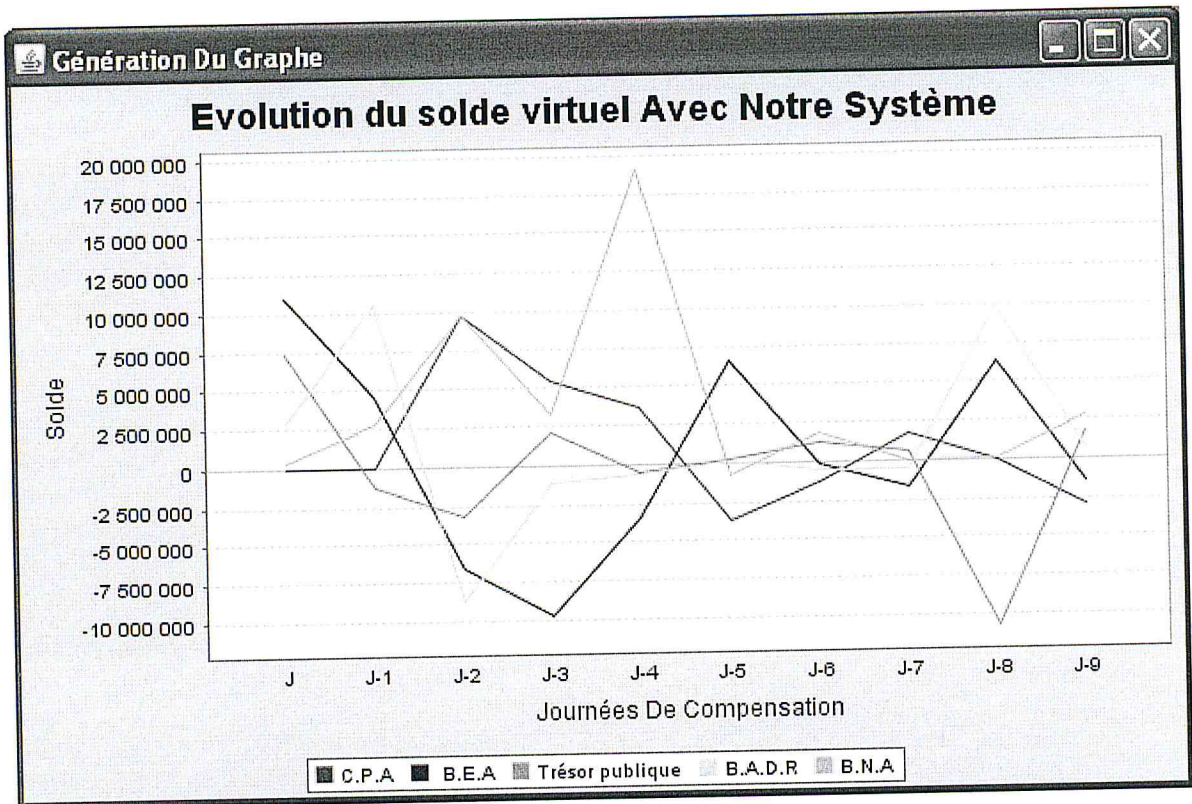


Fig. VII.24 Interface Evolution du solde virtuel Avec Notre Système.

5.25. Génération du fichier XML

Voici en détail une capture du fichier OUTGO généré à la fin de chaque séance de compensation.

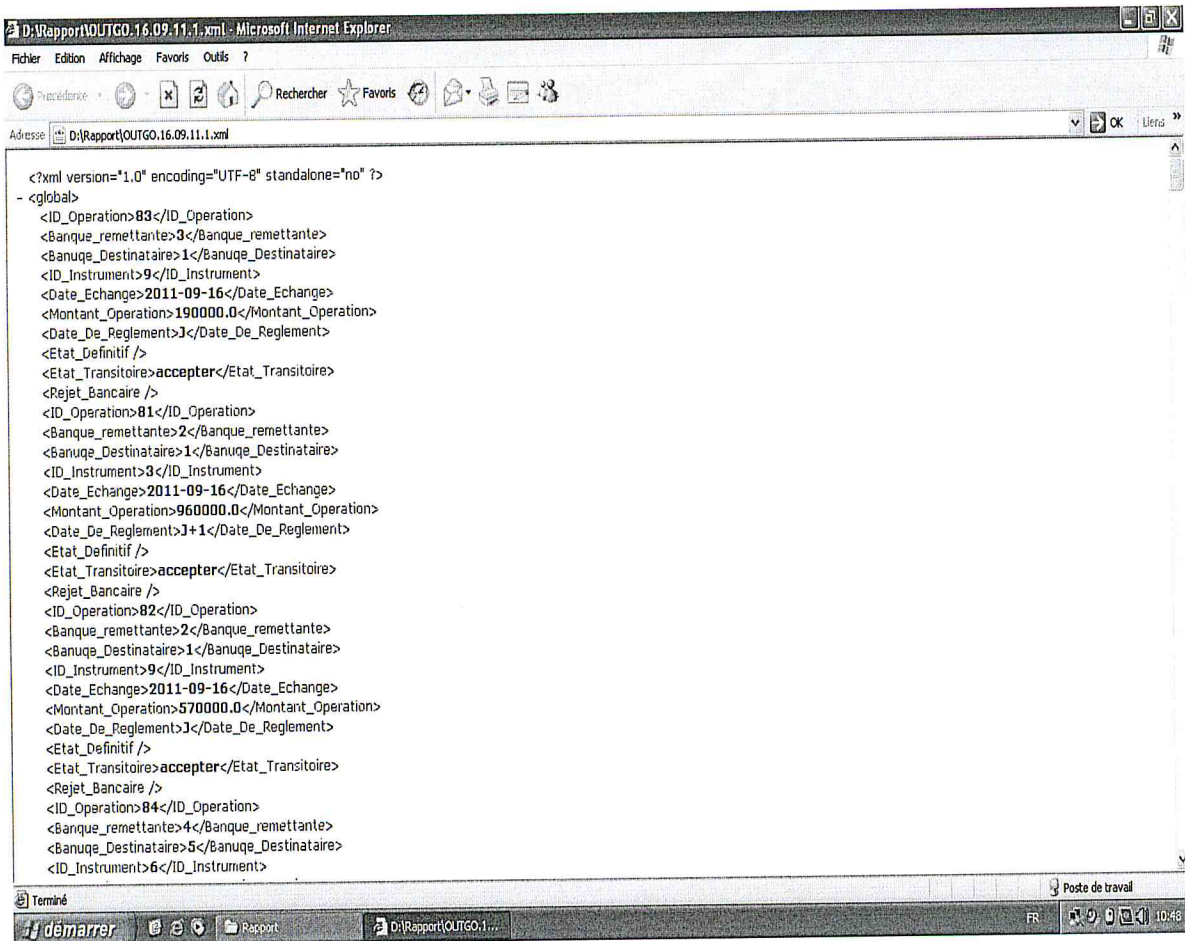


Fig.VII.4.25 Interface Génération du fichier XML.

6. Conclusion

Dans ce chapitre nous avons présenté l'implémentation de notre système ainsi que les résultats de nos expérimentations. Notre système permet d'exécuter toute la chaîne de traitements des opérations de paiements d'une journée à savoir les 3 séances de compensation. En sortie nous avons établis plus graphes de statistiques qui nous permettent de comparer en premier lieu l'évolution des soldes virtuels des banques avec le système actuel et avec notre système.

Conclusion générale

Dans ce mémoire, nous avons conçu et réalisé un système multi-agent de traitement et suivi des opérations de paiement à intégrer au système ATCI.

L'état de l'art nous a permis de voir les différentes caractéristiques des systèmes ATCI et la nécessité d'ajouter une couche supplémentaire de suivi des soldes et de garantir par la suite le bon déroulement des ordres de paiement sans tomber dans des situations d'insolvabilité (défaut de paiement). Les systèmes multi-agents semblent la meilleure approche pour modéliser un tel système, les systèmes de classeurs est l'une des meilleures méthodes d'apprentissage et de raisonnement pour les agents.

Parmi les différents aspects du système ATCI, nous nous sommes intéressés à l'état d'évolution des soldes virtuels des banques, qui est l'une des préoccupations majeures de la banque centrale. L'objectif de notre projet s'est porté donc sur la stabilité des banques financièrement en respectant la contrainte de ne jamais dépasser la limite banque.

Parmi les approches possibles de modélisation qui existent, nous avons donc opté pour l'approche multi-agent car elle permet une meilleure modélisation des différentes composantes du système ATCI. Ainsi nous avons modélisé les banques comme des agents adaptatifs et apprenants, munis d'un module de raisonnement basé sur les systèmes de classeurs afin de leurs permettre d'apprendre de leurs expériences.

Notre système est conçu comme une réponse aux problèmes posés par les risques de défaillance des participants du système de compensation interbancaire, qui oblige l'inversion de la journée de compensation, ce qui implique le recalcul des soldes de compensation des participants non défaillant. Le suivi et le traitement d'une manière intelligente des soldes apparaissent aujourd'hui comme une nécessité pour diminuer les risques d'illiquidités rencontrés durant une séance de compensation. La puissance des applications qui utilisent des Système Multi-Agents laisse à penser que leur place au sein des systèmes d'information ne peut que croître. Cette puissance vient du fait que les systèmes qui utilisent l'approche Multi-Agent essayent d'optimiser les décisions prises selon le domaine

d'application, réduisant ainsi le taux d'erreurs produit par une mauvaise gestion des responsables des banques.

Les travaux menés dans ce mémoire nous ont permis d'approfondir nos connaissances dans le domaine de l'intelligence artificielle (et plus particulièrement les Systèmes Multi-Agents). Notre objectif a été de tirer profit des travaux menés dans cette voie et nous nous sommes intéressées plus particulièrement aux méthodes d'apprentissage, dans le but de développer un nouveau module de suivi Intelligent des soldes.

Les résultats obtenus nous ont montré que les modèles multi-agents peuvent servir à mieux gérer le système ATCI et répondre aux questions des décideurs des ATCI.

En résumé, on dira que les objectifs initiaux ont été totalement satisfaits.

Perspectives :

Notre système répond parfaitement aux objectifs fixés au départ. Néanmoins, nous pouvons citer quelques améliorations qui peuvent rendre notre système plus flexible.

- Créer une interface directe avec le système RTGS pour pouvoir manipuler des soldes réels.
- Ajouter des agents de prédictions qui permettent d'analyser l'historique de façon automatique et intelligente pour établir des prévisions sur l'évolution future du système.

Bibliographie :

Référence	Auteur, œuvre
[ARS, 01]	ARSÈNE SABAS, Systèmes multi-agents : Analyse comparative des méthodologies de développement, mémoire présenté à l'université du Québec à Trois-rivières, Octobre 2001
[BFR, 03]	Banque de France, Revue de la stabilité financière • N°3 • Novembre 2003
[CHA, 04]	Chadare, Rapport de Conception Version Définitive .Atos Euronext / Diamis .Groupe Atos Origin. Version 2.0. 18 juin 2004. http://www.diamis.com/
[CHA,04]	Chadare, Rapport de Conception.ANNEXE 5.Principes Fonctionnels pour la mise en œuvre dans le système d'information. AtosEuronext / Diamis .Groupe Atos Origin.Version 2.0. 18. juin2004. http://www.diamis.com/
[DEM, 95]	DEMAZEAU Y., « From interactions to collective behaviour in agent-based systems », 1995, <i>Proceeding of the First European Conference on Cognitive Science</i> , Saint-Malo, 1995, p. 117-132.
[DJA, 09]	Djama Tania et MOSBAH Samy, Modélisation multi-agent par écoute flottante d'un marché financier, Mémoire de fin d'études en vue d'obtention du diplôme d'Ingénieur en Informatique, Institut National d'Informatique, septembre 2009
[DRO, 93]	A. Drogoul, « De la simulation multi-agents à la résolution collective de problèmes - Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents », Thèse de Doctorat de l'Université Paris VI, Novembre 1993.
[DUV, 01]	C.DUVALLET « <i>des systèmes d'aide à la décision temps réel et distribués : modélisation par agents</i> », Thèse de doctorat, Université du Havre, 2001.
[ERW, 03]	Erwan Livolant, Apprentissage Multi-Agents par Systèmes de Classeurs, Université de Caen, <i>Département Informatique, Laboratoire GREYC</i> , DEA Intelligence Artificielle et Algorithmique, septembre 2003
[FER, 93]	J.Ferber, O.Gutknecht,F.Michel, « From agents to organisations : An organizational view of multi-agent systems”, In Paolo Giorgini, Jörg, P.Müller, and James Odell editors, AOSE, volume 2935 of Lecture Notes in Computer Science, Stanford University, 1993.
[FER, 95]	J. Ferber, <i>Les systèmes multi-agents, vers une intelligence collective</i> . InterEditions, 1995.
[GER, 01]	P. GÉRARD et O. SIGAUD. Généralisation et Apprentissage Latent dans les Systèmes de Classeurs. <i>Extraction des</i>

	<i>connaissances et apprentissage : Apprentissage automatique et évolution artificielle</i> , 3 (1) :87.114, 2001.
[GER, 02]	P. Gerard. Apprentissage de l'anticipation pour la coordination. PhD thesis, Université de Paris VI, 2002.
[GRA, 98]	R. Gray. Agent tcl : A flexible and secure mobile-agent systems. Technical Report PCSTR98-327, Dartmouth College, Computer Science, Hanover, NH, January 1998.
[GÜN, 98]	Güntzer, M – Jungnickel, D – Leclerc, M (1998) Efficient algorithms for the clearing of interbank payments. <i>European Journal of Operational Research</i> 106, 212–219.
[HAC, 08]	Haciane et Akroune « Conception et réalisation d'un Système Multi-Agents pour le commerce électronique », mémoire d'ingénieur d'état en Informatique, Institut National D'Informatique (INI), 2008.
[HAR, 07]	Harry Leinonen (ed.) Simulation studies of liquidity needs, risks and efficiency in payment networks. 2007. Proceedings from the Bank of Finland Payment and Settlement System Seminars 2005–2006. 320 p. ISBN 978-952-462-360-5, print; ISBN 978-952-462-361-2, online.
[HAY, 98]	B. Hayes-Roth, G. Ball, G Lisette, R. Picard, and A. Stern. Affect and emotion in the user interface. In <i>Proceedings of the International Conference on Intelligent user Interface (IUI-98)</i> . ACM Press, 1998.
[JEN, 98]	N. R. Jennings, M. Wooldridge, and K. Sycara A roadmap of agent research and development. <i>Int Journal of Autonomous Agents and Multi-Agent Systems</i> , 1(1):7- 38, 1998.
[JOH, 87]	John H. Holland. Genetic algorithms and classifier systems : Foundations and future directions. In J. J. Grefenstette, editor, <i>Proceedings of the second international conference on genetic algorithms and their applications</i> , pages 82-89, Hillsdale, New Jersey, 1987. Lawrence Erlbaum Associates.
[KIN, 96]	Kinny David, Georgeff Michael, and Rao Anand, A methodology and modeling technique for systems of BDI agents. In W. Van der Velde and J. Perram, editors, <i>Agents Breaking Away : Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'96</i> , (LNAI Volume1038). Springer-Verlag : Heidelberg, Germany,1996.
[KOZ, 2000]	J. Kozlak, « Maintien de l'intégrité fonctionnelle dans les systèmes multi-agents ouverts à ressources renouvelables », Thèse de Doctorat, INP Grenoble, Laboratoire Leibniz, août 2000.
[LEE, 80]	Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, D. Raj Reddy, « The Hearsay-II speech-understanding system: integrating

	knowledge to resolve uncertainty », <i>ACM computing surveys</i> , 12(2) juin 1980, pp. 213-253.
[STO, 98]	Stolzmann W., « Anticipatory classifier systems », <i>Third Annual Genetic Programming Conference</i> , Morgan Kaufmann, p. 658-664, 1998.
[TAI, 04]	Taibi Ahmed, Modernisation De L'infrastructure Du Système Bancaire, rapport de conception, version2.0, 18 juin2004.
[TAI,06]	Taibi, BDL_SYSTEME_TELECOMPENSATION un rapport de banque de la BDL, mai 2006
[THE, 04]	Théophile Gonos, <i>Apprentissage de la locomotion par système de classeurs pour les atomes robotiques du projet MAAM, DEA de Robotique et Systèmes Intelligents (RESIN) Mémoire de Stage, Université Pierre et marie Curie, Juin 2004</i>
[WIL, 94]	Wilson S.W., « ZCS : A zeroth level classifier system», <i>Evolutionary Computation</i> , vol. 2, n° 1, p. 1-18,1994.
[WIL, 95]	Wilson S.W., « Classifier Fitness Based on Accuracy », <i>Evolutionary Computation</i> , vol. 3, n° 2, p. 149- 175, 1995.