



HAL
open science

L'approche multi-agents pour le pilotage des systèmes complexes appliquée aux systèmes du trafic urbain

Ahmad Ali

► **To cite this version:**

Ahmad Ali. L'approche multi-agents pour le pilotage des systèmes complexes appliquée aux systèmes du trafic urbain. Système multi-agents [cs.MA]. Université Blaise Pascal - Clermont-Ferrand II, 2009. Français. NNT : 2009CLF21944 . tel-00725327

HAL Id: tel-00725327

<https://theses.hal.science/tel-00725327>

Submitted on 24 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE BLAISE PASCAL – CLERMONT II
ECOLE DOCTORALE DES SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND

THESE

Présentée par

Ahmad ALI

DEA : Informatique, productique, imagerie médicale

en vue d'obtenir le grade de

DOCTEUR D'UNIVERSITE

Spécialité : INFORMATIQUE

L'approche multi-agents pour le pilotage des systèmes complexes

Appliquée aux Systèmes du Trafic Urbain

Soutenue publiquement le 07/07/2009 devant le jury :

CHABROL Michelle	Co-directrice de thèse
FLORY André	Rapporteur
GOURGAND Michel	Directeur de thèse
ITMI Mhamed	Rapporteur
QUILLIOT Alain	Examinateur
SIMONET Michel	Examinateur

Table des matières :

INTRODUCTION GÉNÉRALE	1
1. CHAPITRE 1 - DOMAINE ET PROBLÉMATIQUE.....	5
1- INTRODUCTION	7
2- LES SYSTÈMES DE TRAFIC URBAIN	8
2-1- <i>Description générale.....</i>	<i>8</i>
2-2- <i>Le système de pilotage</i>	<i>10</i>
2-2-1- <i>L'application Carrefour A Feux CAF.....</i>	<i>16</i>
2-2-2- <i>L'application Mesure De Trafic MDT.....</i>	<i>17</i>
2-2-3- <i>L'application Véhicule de Transport en Commun VTC.....</i>	<i>18</i>
2-2-4- <i>L'application Panneaux à Messages Variables PMV</i>	<i>19</i>
2-3- <i>Problématique</i>	<i>20</i>
3- LA MODÉLISATION DES STU.....	22
3-1- <i>La modélisation de structure et d'écoulement du trafic.....</i>	<i>24</i>
3-2- <i>La modélisation du système de pilotage</i>	<i>25</i>
4- SYNTHÈSE	27
2. CHAPITRE 2 - LE PILOTAGE DES SYSTÈMES DE TRAFIC URBAIN : ÉTAT DE L'ART	29
1- INTRODUCTION	31
2- SYSTÈMES TEMPS RÉEL D'AIDE À L'EXPLOITATION (SAE)	31
2-1- <i>TRANSYT</i>	<i>32</i>
2-2- <i>SCOOT.....</i>	<i>32</i>
2-3- <i>CLAIRE :.....</i>	<i>33</i>
3- SYSTÈMES D'AIDE À LA DÉCISION (SAD).....	33
3-1- <i>TRYS (Cuenca et al, 1995).....</i>	<i>34</i>
3-1-1- <i>Le modèle de l'agent :.....</i>	<i>35</i>
3-1-2- <i>Le modèle du coordinateur :.....</i>	<i>36</i>
3-1-3- <i>KSM.....</i>	<i>36</i>
3-1-3-1- Le langage Concel :.....	37
3-1-3-2- <i>Le langage Link</i>	<i>38</i>
3-3- <i>InTRYS et TRYS A2 (Hernandez et al, 2002).....</i>	<i>39</i>
3-3-1- <i>Les agents de pilotage de trafic</i>	<i>39</i>
3-3-2- <i>Modèles de coordination</i>	<i>40</i>

3-4- (<i>Ossowski et al, 2005</i>).....	41
3-5- <i>CMTMRGS (Adler et Blue, 2002)</i>	43
3-6- <i>Divers:</i>	44
4- SIMULATEURS :	45
4-1- <i>TRANSIMS</i>	45
4-2- <i>AIMSUN2</i>	45
4-2-1- <i>GETRAM</i>	46
4-2-2- <i>EMME/2</i>	46
5- CADRE D'ÉVALUATION.....	46
5-1- <i>Le problème pris en compte</i>	47
5-2- <i>L'approche utilisée pour résoudre chaque problème</i>	48
5-3- <i>Evaluation des résultats</i>	49
6- CONCLUSION	51
3. CHAPITRE 3 - L'APPROCHE MULTI-AGENTS	53
INTRODUCTION.....	55
1- CONTEXTE.....	55
2- CONCEPTS	56
2-1- <i>Agent vs Objet:</i>	58
2-2- <i>Agents vs système expert</i>	58
3- PROPRIÉTÉS ET TYPES D'AGENTS.....	59
3-1- <i>Agent réactif</i>	59
3-2- <i>Agent cognitif</i>	59
3-3- <i>Agent hybride</i>	60
4- MÉTHODOLOGIES BASÉES SUR LES SMA.....	60
4-1- <i>Gaia</i>	60
4-2- <i>Tropos</i>	62
4-3- <i>MaSE</i>	63
4-3-1- <i>La phase d'analyse</i>	64
4-3-2- <i>La phase de conception :</i>	67
4-3-3- <i>AgentTool</i>	68
4-4- <i>Discussion et comparaisons des méthodologies</i>	68
5- OUTILS ET PLATEFORMES POUR LES SMA.....	69
5-1- <i>L'approche AUML</i>	69
5-2- <i>FIPA et la plateforme JADE</i>	70
5-3- <i>La plateforme MadKit</i>	71

6- INTERACTIONS, COMMUNICATIONS ET COORDINATION	76
7- CONCLUSION	77
4. CHAPITRE 4 - PROPOSITION D'UNE APPROCHE AGENT POUR MODÉLISER LE SOUS-SYSTÈME DÉCISIONNEL DES STU.....	79
1- INTRODUCTION	81
2- ASCI-MI UNE MÉTHODOLOGIE MULTIPLE ET INCRÉMENTIELLE POUR LES STU	81
3- L'APPROCHE MULTI-AGENTS DANS LE CONTEXTE DU SOUS-SYSTÈME DÉCISIONNEL SSD	92
4- MASE POUR LA CONCEPTION DU SSD	93
4-1- Analyse	94
4-1-1- Identifier les buts	94
4-1-2- Le modèle de rôles	100
4-2- Conception	102
4-2-1- Construire le modèle de classes d'agents.....	103
4-2-2- Construire les interactions et les communications	104
4-2-3- Les modèle de plan d'agents	106
5- CONCLUSION	107
5. CHAPITRE 5 -CONCEPTION ET IMPLÉMENTATION : UNE ARCHITECTURE GLOBALE D'UN SYSTÈME MULTI-AGENTS POUR LE SSD.....	109
1- INTRODUCTION	111
2- FORMALISER LA CONNAISSANCE DU MODÈLE	112
3- LE MODÈLE MULTI-AGENTS :.....	116
3-1- Madkit et son modèle d'organisation Alaadin-AGR	116
3-2- Passage de la conception d'agent à leur implémentation :	117
4-MODÈLE DE RÉSULTATS	125
5- CONCLUSION	126
6. CHAPITRE 6 - MISE EN ŒUVRE.....	129
1- INTRODUCTION	131
2- CAS D'ÉTUDE : RIOM	132
2-1- Objectifs et problèmes	133
2-2- Modèle de connaissance	133
2-3- Modèle d'action :	135
2-4- Modèle de résultats :	136
2-4-1- Mode 1 : Couplage.....	136
2-4-1-1- Période : 7h30 – 9h30:	137

2-4-1-2- Période : 17h00 – 19h00:.....	142
2-4-2- Mode 2 : interactif.....	146
3- CONCLUSION	147
CONCLUSION GÉNÉRALE.....	149
ORIGINALITÉ DU TRAVAIL	149
APPORTS SPÉCIFIQUES.....	150
PERSPECTIVES	151
RÉFÉRENCES BIBLIOGRAPHIQUES.....	153
ANNEXE A : DÉFINITIONS ET TERMINOLOGIES.....	159
ANNEXE B : GLOSSAIRE DES ACRONYMES.....	171
ANNEXE C : DES STRATÉGIES POUR DONNER LA PRIORITÉ AUX VTC :	173
ANNEXE D : DIAGRAMMES SÉQUENTIELLES DES PLANS DE FEUX SUR LES CARREFOURS DU RIOM	177

LISTE DES FIGURES :

Figure 1-1 : Schéma de la place Delille de Clermont-Ferrand	9
Figure 1-2 : Un schéma de l'infrastructure d'un carrefour	10
Figure 1-3 : Exemple d'un carrefour à 4 branches, 14 mouvements et 8 courants (4 courants de véhicules et 4 courants de piétons).....	11
Figure 1-4 : exemple de diagramme séquentiel pour un cycle en deux phases.	12
Figure 1-5 : Schéma de principe d'un système de pilotage de STU.....	14
Figure 1-6 : Cible générale d'un environnement de modélisation.....	24
Figure 1-7 : Un système intelligent de pilotage de trafic	26
Figure 1-8 : interaction entre le décideur et un système d'aide à la décision.....	27
Figure 2-1 : une architecture générale de TRYS.....	36
Figure 2-2 : InTRYS (Centralisé) et TRYS A2 (Décentralisé).....	41
Figure 2-3 : une architecture abstraite d'un système d'aide à la décision (Ossowski <i>et al</i> , 2005).	42
Figure 3-1 : un agent (perception et action sous-systèmes) dans son environnement (Ferber, 1999).	56
Figure 3-2 : La méthodologie Gaia	61
Figure 3-3 : Les phases de MaSE.....	63
Figure 3-4 : une hiérarchie générale des objectifs dans MaSE.....	65
Figure 3-5 : Exemple d'un modèle de rôles dans MaSE.....	66
Figure 3-6 : Première possibilité d'un diagramme de classes d'agents	67
Figure 3-7 : Diagramme de classes d'agents simplifié	67
Figure 3-8 : types de connecteurs dans AUML	69
Figure 3-9 : diagramme de séquence et échange de messages dans AUML	70
Figure 3-10 : Un modèle de référence des plateformes SMA dans FIPA	71
Figure 3-11 : Structure générale de Madkit	72
Figure 3-12 : Hiérarchie des messages standards.....	73
Figure 3-13 : Modèle AGR étendu.....	74
Figure 3-14 : Amorçage de madkit.....	75
Figure 4-1 : La méthodologie de modélisation ASCI-mi	82

Figure 4-2 : Diagramme de séquence présentant les communications entre les trois sous-systèmes	84
Figure 4-3 : L'environnement ASCImi-STU	86
Figure 4-4 : Diagramme de classe du SSL	87
Figure 4-5 : Diagramme de classe du SSP.....	88
Figure 4-6 : diagramme de classe du SSD.....	89
Figure 4-7 : Les communications entre les trois sous-systèmes, SSL: sous-système logique; SSP: sous-système physique; SSD: sous-système décisionnel	91
Figure 4-8 : Structure hiérarchique des objectifs.	97
Figure 4-9 : Structure hiérarchique des objectifs (modèle dynamique)	99
Figure 4-10 : Le modèle de rôle	101
Figure 4-11 : Le modèle de rôle (simplifié).....	102
Figure 4-12 : modèle général d'agents.....	103
Figure 4-13 : modèle d'agents	104
Figure 4-14 : protocole1	105
Figure 4-15 : modèle de plan de l'agent entité de régulation	107
Figure 5-1 : Le Modèle MVC et les différentes interactions entre ses modules.....	111
Figure 5-2 : architecture d'implémentation.	112
Figure 5-3 : Diagramme d'un carrefour.....	113
Figure 5-4 : Extraction quasi-automatisation du <i>ssd.xml</i> à partir de <i>graphe.xml</i> et <i>ssl.xml</i> ...	114
Figure 5-5 : Exemple de contenu quasi-automatisé du fichier <i>ssd.xml</i>	115
Figure 5-6 : Capture d'écran du module : <i>SSD_Design</i>	116
Figure 5-7 : L'architecture Agent, Group, Rôle (AGR).....	117
Figure 5-8 : Cycle de vie d'un agent sous madkit.	117
Figure 5-9 : Diagramme de classe d'agents.....	119
Figure 5-10 : Hiérarchie des entités décisionnelles du SSD.....	119
Figure 5-11 : Algorithme de distribution de données (calcul de charge par courant).	121
Figure 5-12 : Algorithme de détection de saturation (Surveillance).	121
Figure 5-13 : Algorithme de calcul du nouveau plan de feux pour un CAF.	122
Figure 5-14 : Algorithme de calcul du nouveau plan de feux pour un CTRL et CDZ.....	122
Figure 5-15 : Diagramme de séquence de synchronisation des agents ER	123
Figure 5-16 : Mode 1 - Couplage avec le modèle ASCImi-STU.....	124
Figure 5-17 : Mode 2 : Interactif avec le modèle ASCImi-STU.....	124
Figure 5-18 : L'environnement de modélisation ASCImi-STU SMA.	127

Figure 6-1 : Schéma de mise en œuvre.....	131
Figure 6-2 : Schéma de la ville Riom.	132
Figure 6-3 : Schéma du sous-système physique de la ville Riom avec les configurations décisionnelles associées.	134
Figure 6-4 : Nombre et durée des événements pour la période d'étude.....	138
Figure 6-5 : La structure du carrefour N101.....	139
Figure 6-6 : Les plans de feux originaux du carrefour N101.....	140
Figure 6-7 : Nombre et durée des événements pour la période d'étude suite à l'application des nouvelles configurations.....	141
Figure 6-8 : Nombre et durée des événements entre 17h00 et 19h00.....	142
Figure 6-9 : nombre et durée des événements entre 17h00 et 19h00 pour un seuil = 40% et une durée = cycle/2.....	144
Figure 6-10 : Temps d'attente sur A112 avant : 50579, après : 26121, presque le même comptage cumulatif.....	145

LISTE DES TABLEAUX :

Tableau 1-1 : Le modèle d'un plan de feux et un exemple.....	11
Tableau 1-2 : matrice de sécurité pour le carrefour étudié.	13
Tableau 2-1 : le vocabulaire utilisé dans le modèle de trafic dans KSM.	38
Tableau 2-2 : un format général d'une méthode avec le langage <i>Link</i>	38
Tableau 2-3 : Les activités de régulations traitées	47
Tableau 2-4 : La position des applications par rapport à l'horizon temporel et spatial (la granularité).	48
Tableau 2-5 : L'approche utilisée et l'évaluation de résultats.	50
Tableau 3-1 : Les modèles associés aux étapes de la méthodologie MaSE.	64
Tableau 5-1 : description décisionnelle vs description physique et logique.....	114
Tableau 6-1 : Evénements détectés par CTRL5.....	137
Tableau 6-2 : liste de priorité pour CTRL5	137
Tableau 6-3 : Liste de priorité selon les événements entre 7h30 et 9h30 pour une durée critique = cycle/2 et seuil = 70%.	138
Tableau 6-4 : Les plans de feux originaux et les plans de feux calculés pour le carrefour N101.	140
Tableau 6-5 : Comparaison entre le système initial et le système obtenu sur les carrefours modifiés.....	141
Tableau 6-6 : Une comparaison globale entre le système initial et le système obtenu.	142
Tableau 6-7 : Liste de priorités obtenues.....	143
Tableau 6-8 : Comptage cumulatif des courants des carrefours prioritaires	143
Tableau 6-9 : Nouvelle liste de priorité en fonction des événements pour un seuil = 40% et une durée = cycle/2.	144
Tableau 6-10 : Comparaison entre le système initial et le système obtenu.	144
Tableau 6-11 : comptage sur N105	146

Introduction générale

De nos jours, l'étude et la gestion des Systèmes du Trafic Urbain (STU) sont difficiles et coûteuses, surtout avec les moyens différents et nombreux de transport (véhicules privés, bus, tramway, ... etc.). De plus, la construction et les configurations de tels systèmes sont des tâches très complexes. Dans ces systèmes, l'objectif principal est d'améliorer la circulation et d'optimiser la fluidité du trafic en agissant sur le pilotage et sur les règles de fonctionnement.

L'intelligence artificielle distribuée est une approche pour contrôler les systèmes complexes en décomposant et en distribuant la prise de décision. Les systèmes complexes sont décomposés en sous-ensembles en relation. Chaque sous-ensemble est responsable du contrôle de son domaine et de la coordination des activités avec les autres sous-ensembles.

L'approche distribuée du pilotage peut contribuer à la mise en place d'organisations agiles, combinant des logiques de réseaux et de hiérarchie. Cependant, le pilotage non centralisé des STU s'appuie fondamentalement sur les interactions d'entités communicantes, autonomes et capables de prendre des décisions.

Ce travail s'intègre dans le développement de la méthodologie de modélisation ASCImi-STU (Analyse, Spécification, Conception, Implantation – multiple, incrémentielle pour les STU) qui a pour objectif d'apporter une aide à la conception et à la gestion des STU. Cette méthodologie préconise la construction de deux modèles :

- 1- Le modèle générique de connaissance du domaine : ce modèle est décomposé en trois sous-systèmes logique, physique et décisionnel pour en faciliter la construction.
- 2- Le modèle générique de conception contenant des composants logiciels.

Le sous-système décisionnel, (ou système de pilotage,) contient les règles de fonctionnement. Ce sous-système est le plus complexe et le plus évolutif des trois.

La problématique liée à la conception du sous-système décisionnel des STU est la maîtrise de la cohérence spatiale et temporelle des décisions, compte tenu des caractéristiques des systèmes pilotés et de leur environnement. Cette étude consiste en la réalisation d'un outil d'aide à la décision afin d'améliorer la qualité de service aux usagers et pour résoudre des problèmes de planification au niveau stratégique, tactique et opérationnel.

Une approche basée sur les agents permet de concevoir des modèles de pilotage non centralisés et auto-organisés permettant ainsi d'améliorer la réactivité des systèmes de décision ce qui accroît leur autonomie. Le terme "agent" dénote un matériel ou un logiciel qui soutient un contexte de prise de décision autonome. Les agents ne sont pas simplement des objets ou des acteurs, mais également des entités autonomes essayant d'atteindre des buts en agissant sur leur environnement.

Ce mémoire est structuré en six chapitres.

Le **chapitre 1** s'intéresse au domaine des systèmes de trafic urbain du point de vue décisionnel. Nous présentons tout d'abord une description générale, puis nous abordons le système du pilotage et la problématique. Une description fonctionnelle de ces systèmes est réalisée. Ensuite nous montrons l'intérêt de la modélisation et de la simulation comme outils pour étudier le fonctionnement de systèmes complexes et évaluer les effets des décisions prises.

Le **chapitre 2** réalise un état de l'art sur le pilotage des systèmes de trafic urbain. Deux catégories d'approches sont considérées, la première concerne les systèmes temps réel d'aide à l'exploitation et la deuxième concerne les modèles de pilotage et les systèmes d'aide à la décision. Nous présentons également certains modèles de simulation existant dans la littérature.

Le **chapitre 3** présente les notions et les concepts de l'approche multi-agents. Nous parlons de certaines méthodologies génériques. Quelques outils et plateformes sont également présentés. L'étude réalisée dans ce chapitre permet de bien choisir les méthodes et les outils adaptés pour les aspects décisionnels afin de réaliser un modèle de pilotage des STU.

Dans le **chapitre 4** nous passons à l'utilisation des méthodes et outils choisis précédemment. Nous utilisons la méthode de conception (MaSE) qui est basée sur l'approche agent dans le cadre de la méthodologie de modélisation et de simulation ASCImi-STU. Ce chapitre est dédié à la conception de la partie décisionnelle des STU. La méthode MaSE comporte deux phases : une phase d'analyse afin d'identifier et de structurer les objectifs et une phase de conception permettant de construire le modèle de classes d'agents.

Le **chapitre 5** présente la phase d'implémentation du modèle proposé. Nous utilisons le langage Java sur la plateforme d'implémentation des SMA (MadKit). L'avantage essentiel de cette plateforme est qu'elle est basée sur le paradigme AGR qui permet de représenter la distribution géographique des STU et la hiérarchie des entités décisionnelles. Dans ce

chapitre, nous présentons la construction des entités décisionnelles (actives et passives) sous la forme d'une bibliothèque de composants logiciels. Une entité active est représentée par un « agent » et les entités passives seront modélisées en utilisant une approche orientée objet.

Dans le **chapitre 6** nous traitons un exemple à l'aide de l'environnement construit. Nous nous sommes intéressés à la ville de Riom, pour laquelle nous disposons d'un modèle physique et logique. L'objectif principal est de simuler la prise de décision à plusieurs niveaux hiérarchiques.

En conclusion, nous résumons les principales contributions de notre travail avant d'exposer quelques perspectives envisagées.

Chapitre 1

- Domaine et problématique

Sommaire

1- INTRODUCTION	7
2- LES SYSTÈMES DE TRAFIC URBAIN	8
2-1- <i>Description générale</i>	8
2-2- <i>Le système de pilotage</i>	10
2-3- <i>Problématique</i>	20
3- LA MODÉLISATION DES STU	22
3-1- <i>La modélisation de structure et d'écoulement du trafic</i>	24
3-2- <i>La modélisation du système de pilotage</i>	25
4- SYNTHÈSE	27

1- Introduction

La complexité des systèmes de trafic urbain augmente et les usagers sont de plus en plus nombreux, de nouveaux moyens de transport s'ajoutent au système et partagent le même réseau. La capacité de ce réseau reste par contre limitée et il n'est pas toujours possible de l'élargir.

Afin d'améliorer la performance des Systèmes du Trafic Urbain (STU), il est nécessaire de comprendre leur fonctionnement, et surtout l'ensemble des règles de fonctionnement. Les usagers et les services techniques n'ont pas les mêmes objectifs :

- En ce qui concerne les usagers, leurs objectifs sont de : choisir la direction aux carrefours, minimiser le temps de parcours, réduire le temps d'attente aux carrefours, minimiser le coût, et quelques fois prendre le plus court chemin pour arriver à leurs destinations.
- Pour les services techniques des agglomérations, il s'agit de maintenir une vitesse moyenne d'écoulement raisonnable, de minimiser la saturation, de minimiser le temps d'attente aux carrefours, d'assurer la sécurité ...

Les éléments principaux des STU, les véhicules, parcourent de longues distances et interagissent avec les éléments de contrôle (les feux, les panneaux,...). Les STU sont composés de zones géographiques adjacentes et non isolées. L'ensemble des informations et des décisions est distribué géographiquement et change rapidement au fil de temps. Dans ces conditions, la modification des règles de fonctionnement et la prise de décision dans une zone vont influencer le comportement du système dans les zones adjacentes. Une bonne décision dans une partie ne sera plus bonne peu de temps après. La problématique liée à la conception du pilotage des STU est la maîtrise de la cohérence spatiale et temporelle des décisions, compte tenu des caractéristiques des systèmes pilotés et de leur environnement.

Le problème du trafic urbain se traduit en général par une demande qui est supérieure à l'offre. D'une part, les usagers veulent utiliser le réseau pour atteindre une « destination » à partir d'une « origine » dans les meilleures conditions de trajet et dans un délai raisonnable, d'autre part, la capacité du réseau est limitée et n'est pas suffisante pour un grand nombre des véhicules.

Le contrôle des STU concerne principalement le contrôle des feux de trafic, qui influencent fortement le comportement des véhicules, et également les panneaux à messages variables (PMV) avec lesquels on peut fournir aux usagers des informations importantes concernant l'état actuel du réseau. Cependant ces panneaux n'imposent pas d'obligations aux usagers. D'autres moyens de diffusion de l'état du système existent, comme la diffusion radio et les journaux.

2- Les systèmes de trafic urbain

2-1- Description générale

Un Système de Trafic Urbain est composé des ensembles suivants:

- Des **entités** (les véhicules) parcourant le système en respectant les règles de fonctionnement; ces entités sont des moyens de transport individuel (véhicules particuliers) ou collectifs (transport en commun : les bus, les tramways, ... etc.).
- Un **réseau**, qui est l'ensemble des infrastructures permettant aux entités de se déplacer et comprenant un ensemble d'axes de circulation et de carrefours.
- Des **règles de fonctionnement** imposées par le code de la route et la législation, y compris tous types de signalisation fixe (indicateurs et panneaux) et variable (les feux tricolores et les panneaux à messages variables).

Les STU sont des systèmes complexes¹ de grande taille. Si on prend la place Delille de Clermont-Ferrand comme exemple (voir figure 1-1), cette place est un point crucial pour la circulation dans la ville car plusieurs grands axes se rejoignent et partent de cette place. Certaines voies sont réservées au transport en commun ou aux véhicules prioritaires, et les autres peuvent être empruntées par n'importe quel véhicule. La place est décomposée en deux parties, Nord et Sud, les deux parties contiennent 10 voies d'entrées, 10 voies de sorties, 25 lignes de feux, 62 itinéraires et 45 priorités. Trois ensembles de plans de feux sont utilisés pour sept périodes. Cycle de base 65 secondes pour les périodes 9-11h et 14-17h; cycle entrant de 90 secondes pour les périodes 7-9h et 13-14h; et cycle sortant de 90 secondes pour les périodes 11-13h et 17-19h.

¹ Un **système complexe** est un ensemble composé de nombreux éléments ayant de fortes interactions entre eux et avec l'extérieur, le fonctionnement de ce type de systèmes est difficile à comprendre, et ne peut pas se réduire à la somme de ses éléments « le tout est plus que la somme de ses parties ».

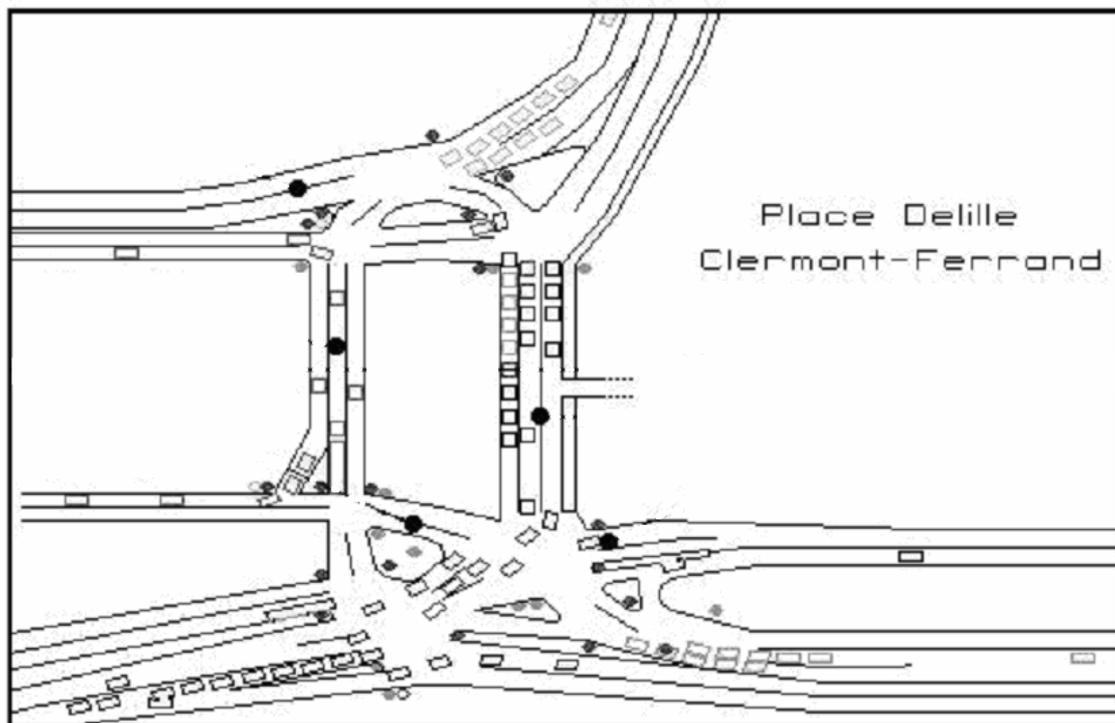


Figure 1-1 : Schéma de la place Delille de Clermont-Ferrand

Il est important pour les services techniques de connaître l'impact de modifications des plans de feux sur la circulation et de bien choisir les bonnes configurations de ces plans pour les différentes périodes. Il est aussi important d'étudier l'efficacité de l'utilisation des plans de feux adaptatifs.

La difficulté de pilotage des STU vient de la présence de flux de types divers (véhicules de transport en commun et véhicules privés), qui partagent le même réseau. Garder un équilibre entre les différents axes sur un carrefour est une tâche complexe.

L'étude liée au pilotage des STU est de type opérationnel et tactique où des modifications en temps réel sont requises sur les signalisations et les plans de feux.

Le schéma de la figure 1-2 présente l'infrastructure d'un carrefour. Dans cette infrastructure on peut trouver : les signalisations (les indicateurs) (1) permettant aux usagers de connaître les règles à appliquer dans les zones à venir. Cette signalisation peut être fixe ou variable. Parmi ces signalisations on trouve les panneaux fixes indiquant la vitesse ou les priorités. Les feux (pour les véhicules, les piétons, les bus, les trams, ...) (2) et les panneaux à messages variables (3) font partie de la signalisation variable ou dynamique.

Les capteurs (4) font partie de l'infrastructure et ont pour objectif de détecter le passage des véhicules et de transmettre des informations aux contrôleurs (5) via des détecteurs.

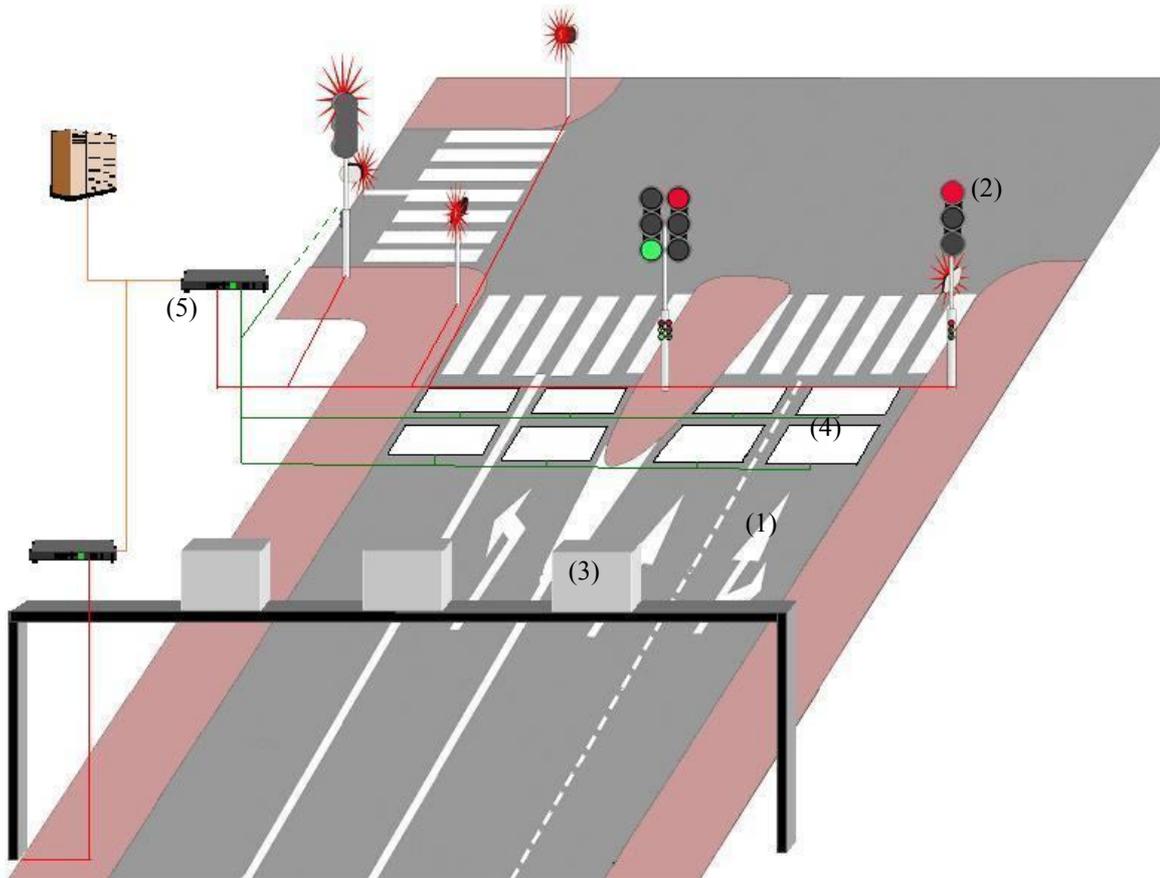


Figure 1-2 : Un schéma de l'infrastructure d'un carrefour

2-2- Le système de pilotage

Avant de présenter le système de pilotage nous présentons quelques termes et définitions dans les STU :

La figure 1-3 montre un exemple d'un carrefour à feux à quatre branches bidirectionnelles. Sur un carrefour, on définit un **courant de circulation** par les mouvements qui ont la même origine et qui utilisent le même **plan de feux**. Il est possible de regrouper plusieurs mouvements de véhicules sur une seule voie, il existe trois types de mouvement : Tourner A Droit (TAD), Tourner A Gauche (TAG) et Tour Droit (TD).

Sur la figure 1-3, on trouve quatre courants de véhicules (F1, F2, F3 et F4) et quatre courants de piétons (P1, P2, P3 et P4).

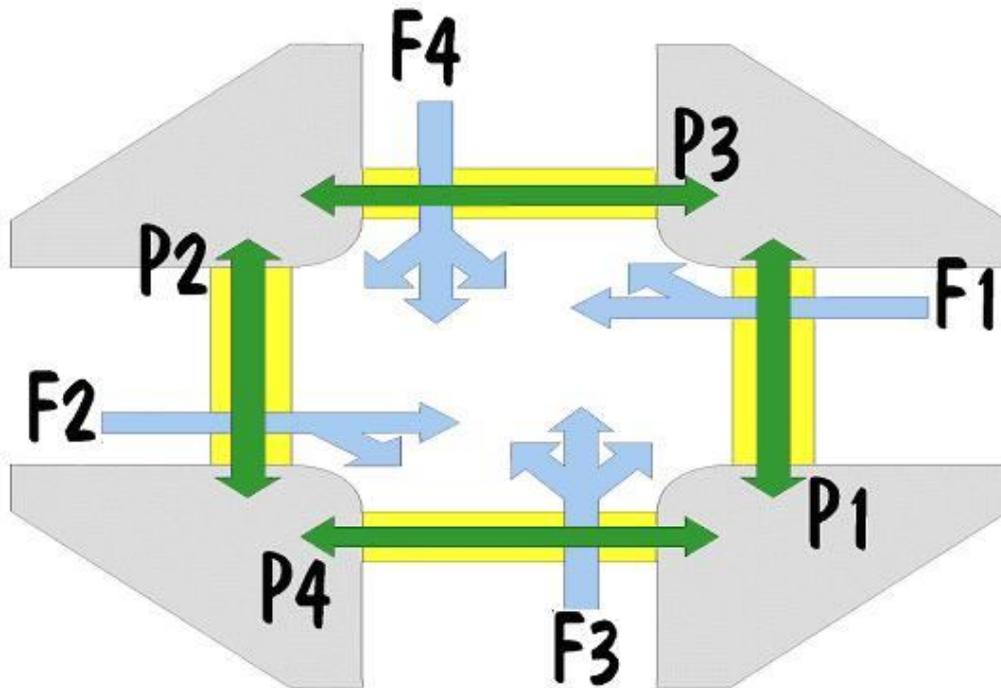


Figure 1-3 : Exemple d'un carrefour à 4 branches, 14 mouvements et 8 courants (4 courants de véhicules et 4 courants de piétons).

Les plans de feux sont appliqués aux courants par des **lignes de feux**, La durée pendant laquelle un ou plusieurs courants sont admis simultanément dans le carrefour constitue une **phase**.

Actuellement, les plans de feux utilisés sont sous forme de fichiers de données d'entrée. La description du format du fichier (format texte) de données plan de feux est la suivante :

Modèle de plan de feux :	Exemple :
Nb de changements de couleurs	4
Durée totale du cycle (en s)	58
Couleur1	Rouge
Fin couleur 1	29
Couleur2	Vert
Fin couleur 2	49
.	Jaune
.	52
CouleurNb	Rouge
Fin couleur Nb	55

Tableau 1-1 : Le modèle d'un plan de feux et un exemple.

Pour l'ensemble de plans de feux utilisés sur ce carrefour, on obtient le diagramme séquentiel suivant (figure 1-4), il y a 8 plans de feux qui correspondent à 8 lignes de

feux, mais dans le reste de ce travail on va considérer que les lignes de feux qui ont le même plan de feux constituent une seule ligne de feux.

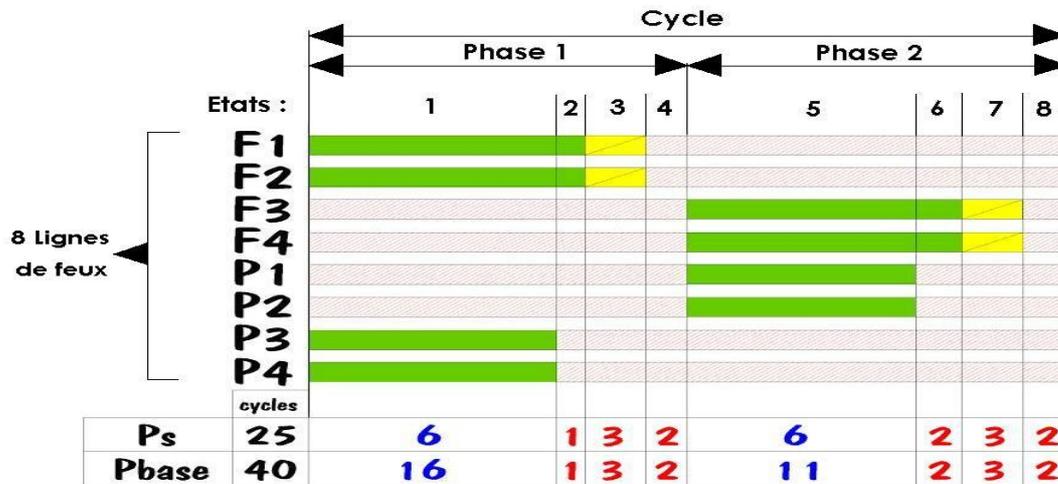


Figure 1-4 : Exemple de diagramme séquentiel pour un cycle en deux phases.

Ce diagramme fait apparaître deux plans de feux. Ps : Plan de feux de Sécurité - P base : Plan de feux de Base. Les plages bleues sont dilatables, les plages rouges sont incompressibles.

Pour chaque phase, le choix de l'allumage du vert sur les différentes lignes de feux repose sur la définition des mouvements compatibles ou non. Cela se traduit par la mise au point de la **matrice de sécurité**. Elle est construite à partir des contraintes de sécurité et de fonctionnement définies dans la réglementation. Entre deux courants déclarés incompatibles au regard du phasage retenu, un intervalle de temps, appelé rouge de dégagement, est nécessaire. Il doit permettre à un piéton engagé à la dernière seconde de vert ou à un véhicule engagé à la dernière seconde de jaune fixe, de dégager la zone de conflit avant l'arrivée du courant antagoniste suivant. Ces temps de dégagement sont calculés sur la base d'une vitesse maximale de déplacement pour les véhicules de **10 m/s** et pour les piétons de **1 m/s**.

La matrice de sécurité est une matrice carrée, dans laquelle sont déclarés tous les antagonismes. Elle indique la durée de rouge minimum nécessaire à respecter entre la fin d'un courant (fin du jaune fixe pour un courant de véhicules ou fin du vert fixe pour un courant de piétons) et le début (allumage du vert) de l'autre. La construction de cette matrice est un passage obligé pour la construction de tout plan de feux de carrefour.

Appliquée à notre exemple, nous obtenons la matrice 1-2 suivante :

	F1	F2	F3	F4	P1	P2	P3	P4
F1	X	X	1	2	0	2	X	X
F2	X	X	2	1	2	0	X	X
F3	1	2	X	X	X	X	2	0
F4	2	1	X	X	X	X	0	2
P1	10	8	X	X	X	X	X	X
P2	8	10	X	X	X	X	X	X
P3	X	X	8	10	X	X	X	X
P4	X	X	10	8	X	X	X	X

Tableau 1-2 : matrice de sécurité pour le carrefour étudié.

Le système de pilotage est composé d'un système de gestion et d'un système de contrôle, ces deux systèmes sont généralement regroupés physiquement dans un même lieu :

- Le **système de gestion** décide la politique de trafic à appliquer (en calculant et modifiant les plans de feux); ce système possède une structure hiérarchique décentralisée et est composé des éléments suivants:
 - Un poste central d'exploitation.
 - Des coordinateurs de zones et des contrôleurs.
 - Un réseau liant le poste central, les coordinateurs et les contrôleurs permettant d'échanger des informations et des commandes.
- Un **système de contrôle** reproduit des commandes et des décisions sur certains éléments du réseau (les feux, les panneaux dynamiques) à partir des données fournies par le système de gestion.

La coordination d'ensemble est réalisée depuis le poste central d'exploitation. La tendance actuelle est à la construction de contrôleurs locaux prenant en charge une partie du système. L'architecture générale du système de pilotage est reprise par la Figure 1-5. Le poste central d'exploitation possède des fonctionnalités de recueil et de stockage des informations provenant du réseau. Il gère une bibliothèque de plans de feux, et possède un algorithme lui permettant de choisir à tout moment un plan de feux correspondant à la situation actuelle. Le fonctionnement, en cas de panne de tout ou partie des contrôleurs, est appelé fonctionnement en mode dégradé. Le nombre de feux

à gérer étant conséquent, des frontaux sont généralement employés pour servir de tampons entre le système et le poste central. Les frontaux assurent :

- La transmission entre le réseau (les contrôleurs) et le poste central,
- Le calcul des plans de feux transitoires (lors d'un changement de plan de feux) et l'envoi des plans de feux aux contrôleurs,
- L'acquisition des informations en provenance des contrôleurs et la vérification du bon fonctionnement des différents appareillages.

Cet équipement est conçu de façon modulaire, chaque module pouvant assurer la gestion de plusieurs carrefours. Cette modularité est très importante compte tenu du niveau de sécurité que le système doit conserver, notamment en mode dégradé.

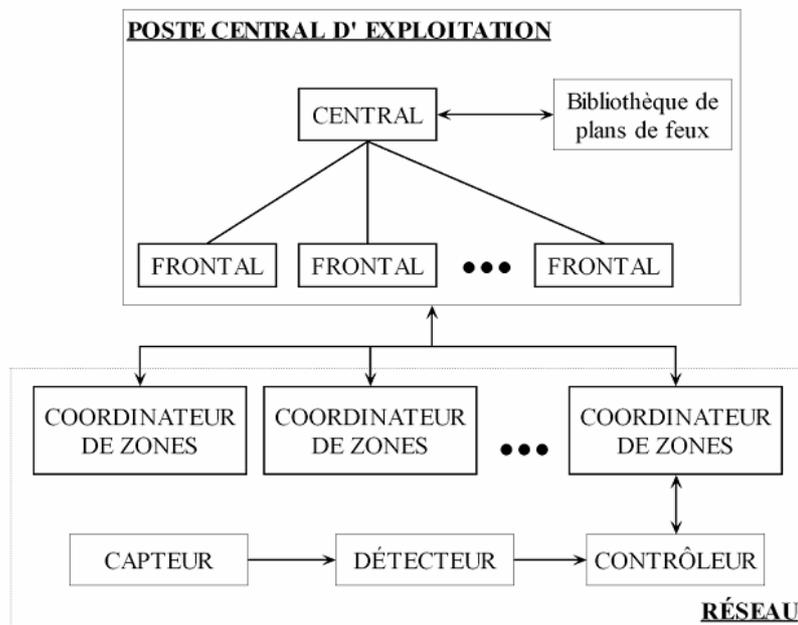


Figure 1-5 : Schéma de principe d'un système de pilotage de STU

L'objectif global des services techniques des agglomérations est de maintenir un niveau raisonnable de performance, d'offrir de bonnes conditions de circulation aux usagers, de maîtriser les conditions de saturation quel que soit l'état du système et d'améliorer la qualité de service (ex : minimiser le temps de parcours, réduire le temps d'attente aux carrefours ...).

Lorsque l'on dispose des comptages des courants sur les intersections, il est relativement facile de calculer les plans de feux au niveau des intersections; plusieurs cas de figure peuvent être distingués :

1- Charges déséquilibrées :

Cette méthode peut être utilisée quand la charge des courants d'une ligne de feux est extrêmement faible. Le feu reste vert dans la direction la plus chargée jusqu'à ce qu'une ou plusieurs voitures approchent du carrefour. Le feu passe alors au vert dans les dernières directions comme nécessaire, jusqu'à un temps maximum prédéterminé. Ce qu'il faut noter c'est que le feu dans les directions les plus chargées ne doit pas rester vert sans interruption même s'il n'y a aucune voiture à l'entrée de l'autre direction. Si c'est le cas, un embouteillage peut résulter aux carrefours adjacents. C'est une des raisons pour lesquelles une stratégie intelligente distribuée et coopérative de commande de trafic est nécessaire.

2- L'onde verte :

Idéalement, les feux doivent être verts quand les voitures arrivent à un carrefour. Une bonne idée est de synchroniser les feux aux carrefours existants sur le même axe d'une façon à permettre aux véhicules de passer les intersections sans attendre. Les véhicules se déplacent en paquets, les feux deviennent verts quand un paquet arrive à un carrefour, et rouge quand il le quitte. Cette méthode permet de diminuer voire supprimer les freinages et les accélérations successifs entre les carrefours concernés. Une telle méthode est difficile à mettre en place sur les deux sens d'un axe, et donc soit un sens est privilégié, soit elle est appliquée alternativement dans un sens puis dans l'autre.

3- Calculer un nouveau plan de feux :

Pour modifier les plans de feux utilisés sur un carrefour, on peut modifier plusieurs variables :

(1) Modifier la longueur de cycle : des heuristiques ont prouvé que l'utilisation des cycles de durée longue augmente la vitesse moyenne de trafic si le flux des véhicules est supérieur à 1300 véhicules/voie/heure (Findler et al, 1992).

(2) Changer des fractionnements de cycle : la durée de vert correspond à la charge des courants.

(3) Changer l'instant de départ de cycle : un instant de départ approprié de cycle réduira le temps d'attente.

4- Véhicules spéciaux :

Certain types de véhicules comme les ambulances, la police,... etc. doivent avoir la priorité dans le réseau de trafic et aux carrefours, le feu de circulation dans leur

direction devrait passer au vert dès que possible, ou rester au vert, indépendamment de la couleur précédente.

De nos jours, des contrôleurs prennent en charge une partie du système : ce sont les carrefours intelligents. Des capteurs et des détecteurs permettent d'obtenir l'état courant qui est ensuite fourni au poste central d'exploitation. Les détecteurs, couplés aux capteurs, traduisent les informations électriques sous la forme d'un débit ou d'une densité.

Des contrôleurs de carrefour assurent le déroulement des plans de feux en cours et la gestion des transmissions vers le poste central d'exploitation et les coordinateurs de zone.

Pour spécifier les fonctionnalités d'une entité de régulation (ER), appelé autrement un centre de décision (CDD), le document de référence est la norme NF P 99-105 (Caractéristiques fonctionnelles des contrôleurs de carrefours à feux).

Selon la norme NF P99-071-1 d'octobre 2001, relative à la spécification du langage DIASER (DIALOGUE Standard pour les Equipements de Régulation), une entité de régulation (ER), qui peut être le poste central, un coordinateur ou un contrôleur, peut gérer une ou plusieurs applications.

2-2-1- L'application Carrefour A Feux CAF

L'application carrefour à feux est l'ensemble des fonctions de commande et de contrôle du fonctionnement d'un ou plusieurs (huit maximum) carrefours à feux.

Les contrôleurs de carrefours à feux fonctionnent en différents modes. Afin de permettre le fonctionnement du contrôleur dans un mode, le contrôleur doit être programmé et câblé en conséquence. La liste des différents modes de fonctionnement est la suivante :

- Initialisation, noté **INI**.
- Jaune clignotant général, noté **JCG**,
- Extinction générale, notée **EXG**
- Mode tricolore, noté **TRI**
 - manuel, noté **MAN**,
 - automatique, noté **AUTO** : local, noté **LOC** ou asservi, noté **ASS**

Une table de correspondance permet de connaître le ou les plans de feux à appliquer dans le mode sélectionné. Le passage d'un mode à un autre est défini et détaillé dans la norme NF P 99-105.

Une **table des commutations**, est utilisée dans les modes (EXG, JCG, LOC et ASS). Elle est constituée des données de commutation de numéros d'ordre différents. La hiérarchie entre les commutations est fixée par ce numéro (priorité au plus grand numéro d'ordre).

A partir de comptages directionnels, la journée est découpée en plusieurs tranches horaires en fonction de l'importance du débit du trafic. A chaque tranche, on associe un plan de feux pour chaque courant du système. Chaque ligne de la table de commutation contient : l'instant de début de la tranche horaire; le mode de fonctionnement dans lequel le contrôleur doit opérer; les indices des plans de feux à appliquer pour les courants du système.

2-2-2- L'application Mesure De Trafic MDT

L'application Mesure De Trafic (MDT) est un ensemble de fonctions d'acquisition locale des données de trafic destinées à la régulation, et des fonctions de contrôle correspondantes.

Les informations transmises permettant l'élaboration par le système externe (un poste central d'exploitation ou un coordinateur de zone est un système externe vis à vis d'un contrôleur qui est une station locale) sont :

- des débits
- des comptages de véhicules
- des taux d'occupation
- la moyenne harmonique des vitesses.

En pratique, les données brutes sont :

- des comptages de véhicules qui permettent l'évaluation des débits :

$$\text{Débit} = \frac{\sum \text{Comptages véhicules}}{\text{Durée mesure}}$$

- des comptages de temps d'occupation qui permettent l'évaluation du taux d'occupation :

$$\text{Taux d'occupation} = \frac{\sum \text{temps d'occupation}}{\text{Durée mesure}}$$

- des comptages des temps de parcours entre deux détecteurs qui permettent l'évaluation des vitesses moyennes :

$$\text{Vitesse moyenne harmonique} = \frac{\sum (\text{comptage véhicules}) \times (\text{longueur mesure})}{\sum (\text{temps de parcours})}$$

On transmet les compteurs bruts et non des quotients comptage/temps. Ces compteurs ne sont pas remis à zéro à l'émission de la donnée. Un comptage pendant une période donnée est obtenu par différence des valeurs enregistrées en fin et en début de la période. Les unités de comptage retenues sont les suivantes :

- Compteurs de véhicules : 1 véhicule
- Compteurs de temps d'occupation : 0,25 seconde
- Compteurs de temps de parcours : 0,25 seconde.

Le langage DIASER met à disposition du programmeur une commande de lecture des mesures de trafic. La station répond par un message contenant le nombre N de compteurs de trafic (N compris entre 0 et 15) suivi de la valeur des compteurs (0 à 255).

2-2-3- L'application Véhicule de Transport en Commun VTC

L'application Véhicule de Transport en Commun (VTC) est un ensemble de fonctions d'identification et de localisation des véhicules de transports en commun et autres véhicules susceptibles de bénéficier du même type de priorité aux feux d'intersection.

Le système externe peut seulement lire les paramètres applicatifs. Cette application est capable de regrouper les informations de 1 à 8 détecteurs répartis sur le site. La lecture des détecteurs actifs par le système externe engendre une réponse par le détecteur VTC du détail des détecteurs actifs : chaque détecteur est symbolisé par une information booléenne significative de la présence d'un véhicule (i.e., 1 bit par détecteur).

Suivant le type de détecteur mis en œuvre, les informations suivantes sont fournies au système externe, à la demande, pour chaque détecteur actif :

- L'identifiant véhicule : identifiant unique du véhicule pour un site donné

- La ligne : identifiant unique de la ligne d'exploitation de VTC pour un site donné
- Le type de véhicule : indique le type (bus, tramway, SAMU, pompiers, autre)
- Le code priorité compris entre 0 et 3 (3 étant le plus prioritaire)
- L'heure de détection du VTC (HHMMSS)
- Si connu :
 - o Le code trajet suivi dans le carrefour
 - o La distance (exprimée en mètres) du VTC à la ligne de feux concernée au moment de la détection
 - o La zone de détection : codage de la zone où le VTC a été détecté
 - o La vitesse du véhicule, exprimée en km/h
 - o L'avance ou le retard, exprimé en minute.

2-2-4- L'application Panneaux à Messages Variables PMV

L'application Panneaux à Messages Variables (PMV) est un ensemble de fonctions de commande et de contrôle du fonctionnement des systèmes d'affichage (le panneau à messages variables, noté PMV);

La liste des commandes détaillées dans DIASER ne couvre pas exhaustivement tous les modes de fonctionnement ni toutes les technologies d'affichage de messages.

Les fonctionnalités portent sur les points suivants :

- Une commande de configuration des paramètres applicatifs du panneau d'affichage
- Une commande de mise en veille du panneau d'affichage,
- Une commande de gestion de la luminosité de l'affichage du panneau d'affichage
- Une commande de contrôle de l'état de fonctionnement du panneau d'affichage ainsi que l'information d'apparition d'un nouveau défaut
- Une commande de déclenchement d'affichage d'un message ASCII
- Une commande de recueil des défauts en cours.

La commande de configuration du panneau :

Cette transaction (écriture des données) permet au système externe de configurer les paramètres applicatifs à prendre en compte par le panneau d'affichage. Les valeurs déterminées par cette transaction portent sur :

- L'unité à prendre en compte pour l'interprétation des valeurs des temps d'affichage (dixième de seconde ou seconde)
- Le temps d'allumage d'un message à afficher en mode clignotant (unité définie ci-dessus)
- Le temps d'extinction d'un message à afficher en mode clignotant (unité définie ci-dessus)
- Les temps d'affichage de chaque message à afficher en mode alterné (unité définie ci-dessus)
- L'unité (dixième de seconde ou seconde ou minute ou heure) à prendre en compte pour l'interprétation de la valeur de temps pour le contrôle de la réception des messages par le panneau (i.e. le time-out)
- Le délai (unité définie ci-dessus) de non réception de message externe pour une mise en veille automatique du panneau (rupture de communication)
- L'unité de temps pour les valeurs des durées de validité d'affichage des messages sur le panneau (dixième de seconde ou seconde ou minute ou heure)
- Le pourcentage de pixels (d'un caractère) H.S. pour la déclaration automatique par le panneau d'un défaut
- Le pourcentage de pixels (d'un caractère) H.S. pour un sabordage automatique du panneau.

Sur réception de la commande de configuration du panneau, le panneau répond par un message d'acquiescement ou par un message d'erreur.

La commande de mise en veille du panneau : Cette transaction permet de déclencher la mise en veille du panneau. A la réception de la commande, le panneau annule l'affichage en cours pour le remplacer par le message associé à la mise en veille. Toute nouvelle commande d'affichage reçue du système externe, alors que le panneau est dans cet état, est prise en compte. Aucune donnée spécifique n'est liée à cette fonction. Sur réception de la commande de mise en veille du panneau, le panneau répond par un message d'acquiescement.

2-3- Problématique

Les systèmes de contrôle du trafic réagissent essentiellement sur les feux de trafic afin de résoudre les problèmes de trafic et d'optimiser la circulation des véhicules. Les problèmes de trafic sont liés à :

- L'identification des zones critiques du système étudié.
- La détection des événements anormaux (saturations, accidents,...etc.)
- La réaction rapide à ces événements afin de garder un état fluide de circulation.

L'efficacité des systèmes de contrôle du trafic dépend des données disponibles sur l'état du réseau, et puisque ces données changent toujours au fil de temps, il faut penser à un système de pilotage qui s'adapte à ces changements, on parle ici des systèmes adaptatifs de contrôle de trafic :

- Dans la première génération des systèmes de contrôle adaptatifs, le système possède une librairie de plans de feux et il choisit le plus convenable pour une situation donnée.
- Dans la deuxième génération : les systèmes utilisent des données réelles du réseau pour produire de nouveaux plans de feux et pour mettre à jour les données historiques afin de les utiliser dans des situations similaires futures.

Qu'est ce qu'une bonne situation de trafic ?

Juger s'il s'agit d'une bonne situation de trafic dans certains cas ne dépend pas de la quantité de véhicules, mais est évalué en fonction de chaque véhicule : peut-il rouler à sa vitesse autorisée sans difficulté ? Peut-on conserver un écoulement avec une vitesse moyenne acceptable ?

Comment résoudre un problème de trafic ?

Le problème le plus fréquent du trafic au niveau opérationnel est le problème de la saturation. Les services de transmission radio essayent de fournir des informations sur des situations qui se trouvent au delà de la vision de l'individu. Ils sont annoncés sur des fréquences spéciales. Ceux-ci peuvent fournir des informations en temps réel aux utilisateurs. Malheureusement, ces services tendent à être lents et fallacieux. Ils fournissent souvent des informations quand il est déjà tard pour réagir convenablement. Dans certains cas, la situation se détériore en raison de l'information ou des réactions inconvenables et exagérées.

Pour résoudre le problème de saturation plus efficacement, les gestionnaires devraient adopter deux étapes : premièrement **choisir un plan de feux** qui doit être appliqué (même s'il reste seulement un plan primitif); deuxièmement **exécuter le plan**.

Les contraintes liées à la structure physique du système et les règles de gestion du trafic doivent être respectées. Le plan initial doit être modifié selon la situation de trafic.

Toute action sur le système de pilotage peut avoir des conséquences imprévisibles sur le trafic et les décideurs sont de plus en plus demandeurs d'outils d'aide à la décision pour évaluer les impacts d'une modification.

3- La modélisation des STU

La simulation est un outil permettant d'évaluer les effets d'une décision pour qu'elle reproduise le comportement du système et montre la manière dont il peut réagir.

Le pilotage agit sur le fonctionnement du système par l'intermédiaire du système de contrôle en modifiant par exemple le phasage de feux. Dans ce contexte on peut réaliser:

- Une étude à priori : le système réel n'existe pas et l'objectif est de déterminer le type et l'emplacement des infrastructures nécessaires à l'écoulement du flux; la simulation permet alors le test d'hypothèses.
- Une étude à posteriori : le système réel existe et des modifications sont à envisager suite à une augmentation de la demande de trafic. L'ensemble du système doit être évalué mais seules des modifications locales sont envisageables.

Une étude relative à la conception et/ou à la modification d'un STU est envisageable à différents niveaux :

- Le niveau **stratégique** sur du long terme, relatif à la gestion et à la maintenance des STU. Introduire de nouveau type de transport, ...

- Le niveau **tactique** sur du moyen terme, concerne les travaux de maintenance saisonnière par exemple.

- Le niveau **opérationnel** sur du court terme : concerne la modification de plans de feux, la prise en compte des incidents.

L'objectif de nos travaux est de proposer, dans un cadre méthodologique, un outil pour l'aide au pilotage des STU permettant de tester différentes techniques de régulation afin de résoudre des problèmes de planification au niveau stratégique, tactique et opérationnel. Nous pouvons différencier deux types de régulation :

- **Macro régulation** désigne le choix et l'enclenchement des différents programmes, au niveau de l'ensemble du système, en fonctions des caractéristiques générales du trafic (heures de pointes, heures creuses, etc.).
- **Micro régulation** désigne les choix effectués à l'intérieur d'un programme, sur un carrefour, en fonctions des caractéristiques particulières du trafic sur ce carrefour.

La modélisation et la simulation sont aujourd'hui largement utilisées car elles permettent de bien comprendre le comportement des systèmes étudiés, le test d'hypothèses et l'anticipation de certains résultats. L'apport et l'efficacité de la simulation sont augmentés si celle-ci est utilisée dans un environnement de développement, d'où la notion d'environnement de modélisation. Un tel environnement correspond à un ensemble de méthodes et d'outils répartis en couches. La figure 1-6 montre une cible générale d'un environnement de modélisation représentant les fonctionnalités qu'un tel environnement doit offrir :

- Un logiciel d'**évaluation des performances** nécessitant une phase de validation s'il repose sur la simulation, car la simulation ne repose sur aucune base mathématique,
- Une couche **Graphique** pour la saisie du modèle conceptuel, pour la spécification du fonctionnement, pour l'exploitation des résultats et pour l'animation,
- Une couche **Aide à la Décision** pour accéder à des outils d'aide à la décision,
- Une couche **Gestion des Données** pour l'accès aux bases de données (suite à des récoltes de statistiques, des enquêtes de ménage, ...),
- Une couche **Recherche Opérationnelle et Statistique** pour exploiter les résultats fournis par les modèles et pour déterminer le meilleur routage, un nouveau phasage...
- Une couche **Méthode d'analyse et de Spécification et Outils de Spécification** contenant des outils pour la description de la structure, des flux présents dans le système et du fonctionnement du système,
- Une couche **Méthodologie de modélisation** du domaine pour la construction de modèles de simulation sans qu'il soit nécessaire de posséder des compétences

dans ce domaine. Le modèle de simulation est alors construit automatiquement à partir du modèle conceptuel.

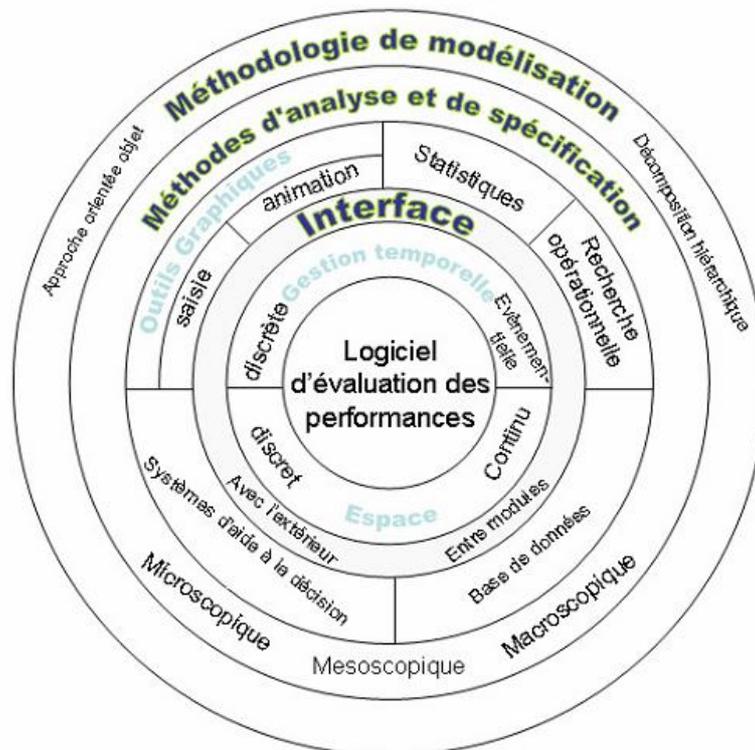


Figure 1-6 : Cible générale d'un environnement de modélisation

3-1- La modélisation de structure et d'écoulement du trafic

Dans le domaine des STU, beaucoup de modèles ont été proposés pour aider les experts du domaine à comprendre certaines situations et à prendre des décisions pour résoudre des problèmes rencontrés.

Les environnements de modélisation actuels correspondent plus à une surcouche de simulateur car ils ne contiennent pas de méthodes d'analyse et de spécification. Ils sont orientés plutôt pour la modélisation de la charge (l'écoulement des véhicules) mais ils ne prennent pas en compte la modélisation du réseau (le sous-système physique) ou de son pilotage (le sous-système décisionnel).

Les travaux présentés par (Sarramia, 2002) utilisent une méthodologie de modélisation des systèmes du trafic urbain qui couvre la majorité des couches nommées précédemment. Cette méthodologie permet de concevoir un modèle de simulation dont la mise en œuvre nécessite la description de l'infrastructure du système étudié et permet la détermination des zones critiques qui génèrent des saturations locales ou globales, et

d'agir sur le système pour maîtriser ou améliorer l'état du système. Les moyens d'action interviennent soit sur l'architecture du réseau, soit sur le pilotage, soit sur les deux. Le modèle donne en sortie un volume important de données (telles que les traces de simulation), et il est difficile même pour un expert du domaine de comprendre le sens de ces données, d'où la nécessité de formaliser la connaissance du domaine du point de vue décisionnel relativement aux besoins des spécialistes et des services techniques.

3-2- La modélisation du système de pilotage

Considérant la complexité du pilotage des STU et les différents niveaux hiérarchiques de prise de décision, une modélisation multi-agents du sous-système décisionnel a été présentée dans les travaux de (Sarramia, 2002); ses travaux ont permis d'identifier les entités passives de ce sous-système (les capteurs, les détecteurs et les feux de circulation) et les entités actives (le poste central, les coordinateurs de zones et les contrôleurs). Par contre, les fonctionnements de ces entités et leurs communications n'ont pas été définis.

Dans ce travail, nous nous concentrons sur le sous-système décisionnel des STU afin de proposer un modèle multi-agents représentant cette partie et permettant de simuler son fonctionnement.

Le modèle générique de connaissance obtenu par (Sarramia, 2002) contient une bibliothèque riche de composants logiciels décrivant les parties physiques et logiques. Modéliser le pilotage des STU permet d'enrichir ce modèle générique de connaissance en ajoutant de nouveaux composants qui représentent les entités décisionnelles et à utiliser ces composants pour modéliser le processus de raisonnement de ces entités afin de résoudre les problèmes rencontrés.

Nous proposons de voir le système de pilotage de trafic (les centres de décision) comme un couplage entre le pilotage de trafic et la modélisation des connaissances (Figure 1-7) :

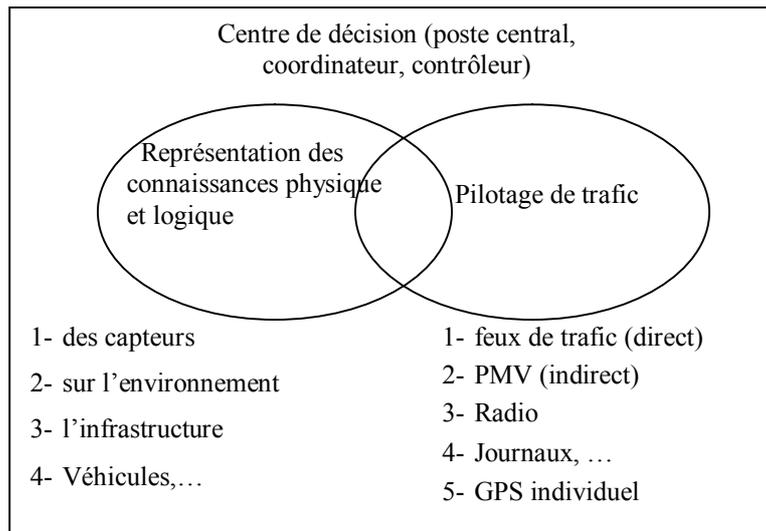


Figure 1-7 : Un système intelligent de pilotage de trafic

Le processus de modélisation des entités de régulation nous amène également à la réalisation d'un outil d'aide à la décision. L'objectif le plus simple et le plus primitif d'un système d'aide à la décision est de fournir au décideur les informations nécessaires concernant le système étudié et c'est le décideur qui prend des décisions à partir de ces données; toutefois les systèmes d'aide à la décision les plus avancés essaient non seulement de fournir des données mais d'analyser ces données et de suggérer des solutions.

Dans les systèmes d'aide à la décision, l'interface utilisateur est une couche importante et nécessaire pour assurer une bonne interaction homme – machine. Traditionnellement, deux modèles extrêmes possibles peuvent être considérés :

- Le modèle qui réduit au minimum le rôle du système. Le système fournit les bonnes informations et l'utilisateur contrôle totalement la prise de décision.

- Le modèle qui réduit au minimum le travail et la responsabilité de l'utilisateur. L'utilisateur est au service de l'application, fournissant simplement des données, et l'application assume automatiquement des décisions.

Un outil d'aide à la décision doit faire des propositions à l'utilisateur et des explications justifiant des questions comme : Pourquoi avez-vous besoin de ces informations ? Comment avez-vous obtenu cette conclusion ? Qu'est-ce qui se produira si je décide : D1, D2, ..., D4 ? Pourquoi cela se produira-il ? Comme le montre la figure 1-8, (Ossowski et al, 2005).

L'utilisateur met en application des options après une conversation utilisateur - système dans les limites des questions précédentes.

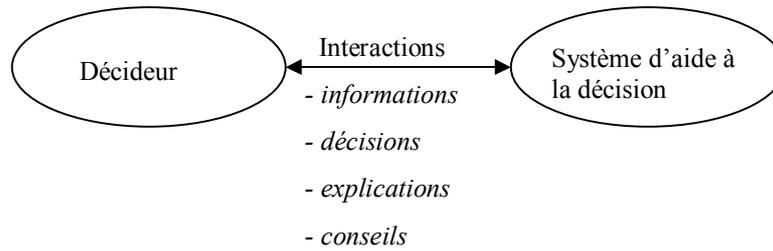


Figure 1-8 : interaction entre le décideur et un système d'aide à la décision.

L'objectif de nos travaux de thèse est de concevoir, réaliser et mettre en œuvre un système efficace d'aide à la décision qui est capable de :

- 1) Identifier des problèmes (information, classification)
- 2) Exprimer les problèmes avec des termes causes - conséquences (explication, diagnostics)
- 3) Prendre des décisions (actions, décisions)
- 4) Prévoir les conséquences des décisions (prévisions, conseils)

4- Synthèse

Pour atteindre notre objectif nous proposons de définir et de mettre en œuvre une approche multi-agents pour modéliser le système de pilotage des systèmes du trafic urbain.

Ces travaux s'intègrent dans la méthodologie de modélisation nommée ASCI (Analyse, Spécification, Conception et Implémentation) (Gourgand et Kellert, 1991). Cette méthodologie a été utilisée et adaptée dans différents domaines : les systèmes industriels de production (Goujon, 1998), (Lacomme, 1998), et (Tchernev, 1997); les systèmes hospitaliers (Combes, 1994), (Fenies, 2006); et les systèmes du trafic urbain (Sarramia, 2002), (Chabrol *et al*, 2001, 2003 ,2005).

Pour étudier les systèmes du trafic urbain, (Sarramia, 2002) a proposé une extension adaptée de cette méthodologie nommée ASCI-mi (m pour multiple, i pour incrémentielle), ses travaux ont permis de modéliser les parties physique et logique des STU, et d'identifier les entités passives et actives de la partie décisionnelle sans en préciser leurs fonctionnements détaillés et leurs communications.

Dans les chapitres suivants nous nous focalisons sur la partie décisionnelle des STU et sur l'utilisation de l'approche multi-agents dans le cadre de la méthodologie ASCI-mi pour conceptualiser et modéliser cette partie.

Chapitre 2

- Le pilotage des systèmes de trafic urbain : état de l'art

Sommaire

1- INTRODUCTION	31
2- SYSTÈMES TEMPS RÉEL D'AIDE À L'EXPLOITATION (SAE)	31
2-1- <i>TRANSYT</i>	32
2-2- <i>SCOOT</i>	32
2-3- <i>CLAIRE</i> :	33
3- SYSTÈMES D'AIDE À LA DÉCISION (SAD)	33
3-1- <i>TRYS</i> (Cuenca et al, 1995).....	34
3-3- <i>InTRYS</i> et <i>TRYS A2</i> (Hernandez et al, 2002).....	39
3-4- (Ossowski et al, 2005).....	41
3-5- <i>CMTMRGS</i> (Adler et Blue, 2002).....	43
3-6- Divers:.....	44
4- SIMULATEURS :	45
4-1- <i>TRANSIMS</i>	45
4-2- <i>AIMSUN2</i>	45
5- CADRE D'ÉVALUATION	46
5-1- <i>Le problème pris en compte</i>	47
5-2- <i>L'approche utilisée pour résoudre chaque problème</i>	48
5-3- <i>Evaluation des résultats</i>	49
6- CONCLUSION	51

1- Introduction

Dans ce chapitre un état de l'art sur le pilotage des STU est présenté. Les applications et les outils concernant le pilotage des systèmes de trafic urbain qui existent dans la littérature peuvent être classés en deux grandes catégories : les **systèmes d'aide à l'exploitation** temps réel ou différé, et les **outils d'aide à la décision**.

Un système d'aide à l'exploitation est un automate ayant un mécanisme autonome pour exécuter une action requise à un moment donné ou suite à une certaine condition (McGraw-Hill, Science and Technology, 2003). L'autonomie du système est sa capacité d'agir et de prendre des décisions.

Un système d'aide à la décision est un système qui fournit à l'utilisateur des informations spécifiques l'aidant à prendre des décisions. Ce type de système a une influence indirecte au niveau de la prise de décision et la décision finale est faite par le décideur.

Dans la première section de ce chapitre nous présentons trois systèmes temps réel d'aide à l'exploitation. Ces systèmes sont installés dans des grandes villes et utilisent des données réelles pour prévoir ou traiter les cas de saturation.

La deuxième section traite les systèmes d'aide à la décision; ce sont des modèles de pilotage et nous nous concentrons essentiellement sur les modèles basés sur une base de connaissance ou sur une approche orientée agents.

Dans la troisième section, quelques modèles de simulation des systèmes du trafic urbain sont présentés; ces modèles sont des outils qui aident les experts du domaine à analyser certaines situations et à comprendre le fonctionnement du système.

Finalement nous décrivons un cadre d'évaluation des outils existants, et nous comparons ces outils selon les critères du cadre défini.

2- Systèmes temps réel d'aide à l'exploitation (SAE)

Ces systèmes sont des automates installés dans des grandes villes qui utilisent des données réelles, ils possèdent un auto-mécanisme pour exécuter une action requise à un instant prédéfini ou suite à l'occurrence de certaines conditions.

2-1- TRANSYT

TRANSYT (Traffic Network Study Tool) est un système qui a été développé par le laboratoire de recherche du trafic (TRL) pour créer une série de plans de feux à temps fixe pour des périodes différentes de la journée ou pour des conditions de trafic prédéfinies; il peut également calculer la durée optimale du cycle pour un problème de congestion donné (Robertson, 1986).

TRANSYT est un logiciel qui ne travaille pas sur des données réelles du réseau, mais il exige des données et des statistiques récoltées et analysées pour chaque situation. A partir de ces données il produit un plan de feux optimal pour une partie de réseau ou pour coordonner plusieurs parties. Le processus d'optimisation d'un plan de feux exige beaucoup de temps, et si les données ne sont pas mises à jour, les plans de feux deviennent moins efficaces au fil du temps.

Les changements fréquents de plans de feux dans le système provoquent des perturbations de circulation sur le réseau urbain.

2-2- SCOOT

SCOOT (Split Cycle Offset Optimisation Technique) est un système qui a été développé par une coopération entre Peek Traffic Ltd., Siemens Traffic Controls et le laboratoire de recherche du trafic (TRL). SCOOT est un système de contrôle adaptatif installé principalement à Londres et dont l'objectif est de surveiller le réseau urbain en permanence. Lors d'une congestion il modifie légèrement les plans de feux pour réduire les délais et améliorer la circulation (après saturation) (Hunt, 1982), (Bretherton, Bowen, 1990).

Les données sont récoltées depuis des détecteurs installés sur les intersections. Les nouvelles versions de SCOOT prennent en compte la priorité pour certains types de véhicules comme les véhicules de transport en commun (bus, tramway, ...).

D'autres systèmes ont été développés par TRL qui sont des extensions de SCOOT, comme ASTRID qui utilise les données de SCOOT pour fournir une base de données historiques du trafic et faire des comparaisons entre différentes situations. INGRID est un autre système qui essaye de détecter les incidents à partir des données de SCOOT.

Il existe d'autres systèmes SCOOT-like comme : SCATS (Lowrie, 1982); OPAC (Gartner, 1983); PRODYN (Farges et al, 1983); UTOPIA (Mauro et al, 1989).

Il est possible d'utiliser des systèmes de supervision qui fonctionnent parallèlement avec SCOOT, comme CLAIRE.

2-3- CLAIRE :

CLAIRE est un système indépendant d'aide à l'exploitation et de supervision de la congestion du trafic, il est parmi les premiers systèmes de gestion de trafic qui ont été fondés sur une base de connaissances et qui ont été testés sur des cas réels. Il a été implémenté et intégré dans le système de contrôle de trafic dans plusieurs villes européennes comme Paris, Toulouse, Rennes, Bruxelles et Londres (son objectif est de prévoir l'état du système avant la saturation) (Scemama, 1992), (Scemama, 2000).

CLAIRE possède plusieurs couches de fonctions permettant d'assurer en temps réel :

- Le diagnostic de la congestion des réseaux et la traduction des données du trafic en indicateurs symboliques : saturé, fluide, inconnu.
- Le contrôle de la congestion et la proposition ainsi que l'implémentation des recommandations en terme d'actions : optimiser les plans de feux.
- L'information sur les formes de la congestion aux usagers.

CLAIRE peut assurer d'autres fonctions en temps différé comme :

- une analyse stratégique des scénarios de congestion.
- une évaluation des performances du réseau.
- une aide à la conception de nouvelles stratégies contre la saturation.

Ce système est fondé sur une approche orientée objet et est écrit en langage C++, il contient une interface Homme-Machine et des moteurs d'inférence pour les règles. Le module d'observation de la congestion possède une interface hypertexte utilisable à partir d'un navigateur Internet. En plus, le système fonctionne sur des plateformes Windows ou Unix et il utilise des protocoles FTP pour transférer les données, ce qui permet une architecture client - serveur simple et robuste. Les travaux en cours concernent la gestion des perturbations des réseaux de transport en commun.

3- Systèmes d'aide à la décision (SAD)

Plusieurs solutions pour un pilotage adaptatif du réseau et des applications basées sur les systèmes multi-agents pour des objectifs de contrôle et d'optimisation ont été suggérées.

Les travaux basés sur la connaissance appliqués à la gestion de trafic a commencé à la fin des années 80 (Scemama, 1989; Cuenca, 1989). En Finlande, L'OECD (OECD, 1990; 1992) a créé un groupe d'experts sur les systèmes experts dans le transport. En outre, le programme européen d'ATT/DRIVE a favorisé différents projets expérimentaux tels que CLAIRE (Scemama, 1992) et KITS (Boero *et al*, 1993, 1994); (Cuenca *et al*, 1992a, b).

Rôle des systèmes basés sur la connaissance dans la gestion de trafic :

Les systèmes de gestion de trafic doivent être réactifs aux différents états de la circulation dans le réseau commandé. Dans les premiers systèmes, l'approche était basée sur une bibliothèque de plans de feux à appliquer en ligne dans différentes situations prédéfinies selon certains critères basés sur le temps (systèmes à temps-fixe) ou sur les données de trafic récoltées par des capteurs au bord de la route (Cuenca *et al*, 1995).

L'expérience de l'utilisation des systèmes comme SCOOT prouve que la bonne performance est atteinte quand les situations de trafic ne sont pas critiques, ou les saturations se produisent seulement pour des intervalles limités de temps. Dans des situations constamment saturées, l'intervention d'un opérateur est habituellement exigée car :

1. L'évaluation de la situation de trafic par le système peut être approximative puisque les capteurs sont parfois insuffisants pour détecter les cas de saturation (par exemple la longueur des files d'attente à la ligne d'arrêt peut déborder après les capteurs placés pour la détection de file d'attente).
2. Pour la stabilité du système dans le réseau, les décisions prises par le système sont de petits réglages sur le plan actuel de feux, alors que des changements plus énergiques pourraient souvent être exigés.

Le même besoin de tenir compte des aspects spécifiques dans des zones problématiques apparaît dans la commande d'autoroute par des panneaux à messages variables. Le conseil affiché aux gestionnaires devrait être suffisant pour :

1. Permettre aux utilisateurs de prendre certaines options disponibles;
2. Ne pas produire de flux excessif en dehors de l'autoroute, qui peut saturer des options (trajets) alternatives. Chaque problème exige donc l'analyse de la situation en utilisant la connaissance des critères de comportement et de commande de trafic spécifiques pour cette zone.

3-1- TRYS (Cuenca *et al*, 1995)

TRYS est un modèle fondé sur une base de connaissances pour un système de pilotage adaptatif du trafic (Cuenca *et al*, 1995) qui a été implémenté dans certaines villes en Espagne. TRYS est un système temps réel qui exige l'intervention de l'homme. On peut considérer que

ce système est le premier système utilisant une base de connaissances, une telle architecture a les avantages suivants :

- 1- La possibilité de modéliser la connaissance là où il n'est pas possible d'exprimer les problèmes sous forme algorithmique.
- 2- Une gestion facile de la connaissance avec une interface familière même pour les utilisateurs non-programmeurs.
- 3- L'obtention des explications des problèmes et des informations récoltées

Puisqu'il est difficile de surveiller le système dans sa totalité, le principe de l'architecture de TRYS est de diviser la zone étudiée en plusieurs parties; chaque partie est gérée par un composant logiciel (agent), et ces agents sont liés à un autre agent de niveau plus haut (un coordinateur). Ce principe permet de bien comprendre les causes et les évolutions des problèmes rencontrés.

3-1-1- Le modèle de l'agent :

L'objectif de l'agent est de fournir deux types d'information : des diagnostics et explications, et des actions de contrôle. La connaissance de l'agent se compose de 3 grandes sections :

- 1- La structure physique : elle comprend une connaissance sur le comportement du trafic dans sa zone, et des moyens de récolter des informations depuis les capteurs; à partir de ces informations l'agent peut calculer des valeurs de base concernant sa partie (vitesse, concentration, fluidité).
- 2- Les problèmes de trafic : l'agent analyse les données requises depuis des détecteurs pour signaler les problèmes et préciser leur gravité : - Où est le problème ? - Quelle est sa gravité ? - Quelle est la cause ?
Les différentes variables qui influencent le problème sont classées comme suit :
 - L'état de contrôle : l'état des flux à l'entrée de la zone de problème, les valeurs des panneaux à message variables qui sont à l'entrée de la zone du problème,
 - Les entrées et les sorties de la zone : contenant les flux des véhicules à l'entrée et la sortie de la zone pour une période suffisante pour qu'un véhicule passe la zone par le plus long trajet,
 - Les valeurs sur la fluidité et la concentration qui sont disponibles,

- Les zones de saturation,
- Les sections critiques de la zone de saturation.

3- Les actions de contrôle : l'ensemble des panneaux à messages variables à utiliser; ces messages contiennent les noms des zones affectées par le problème et leurs états, et les chemins alternatifs que les usagers peuvent utiliser pour éviter la zone à problème.

3-1-2- Le modèle du coordinateur :

L'objectif du coordinateur est de déterminer la compatibilité entre les différentes solutions proposées et de résoudre les conflits entre ces solutions suite à un problème (i.e., d'actions différentes sur la même entité physique); en plus il doit déterminer la priorité des agents qui lui appartiennent afin de résoudre les conflits.

La figure 2-1 résume cette architecture :

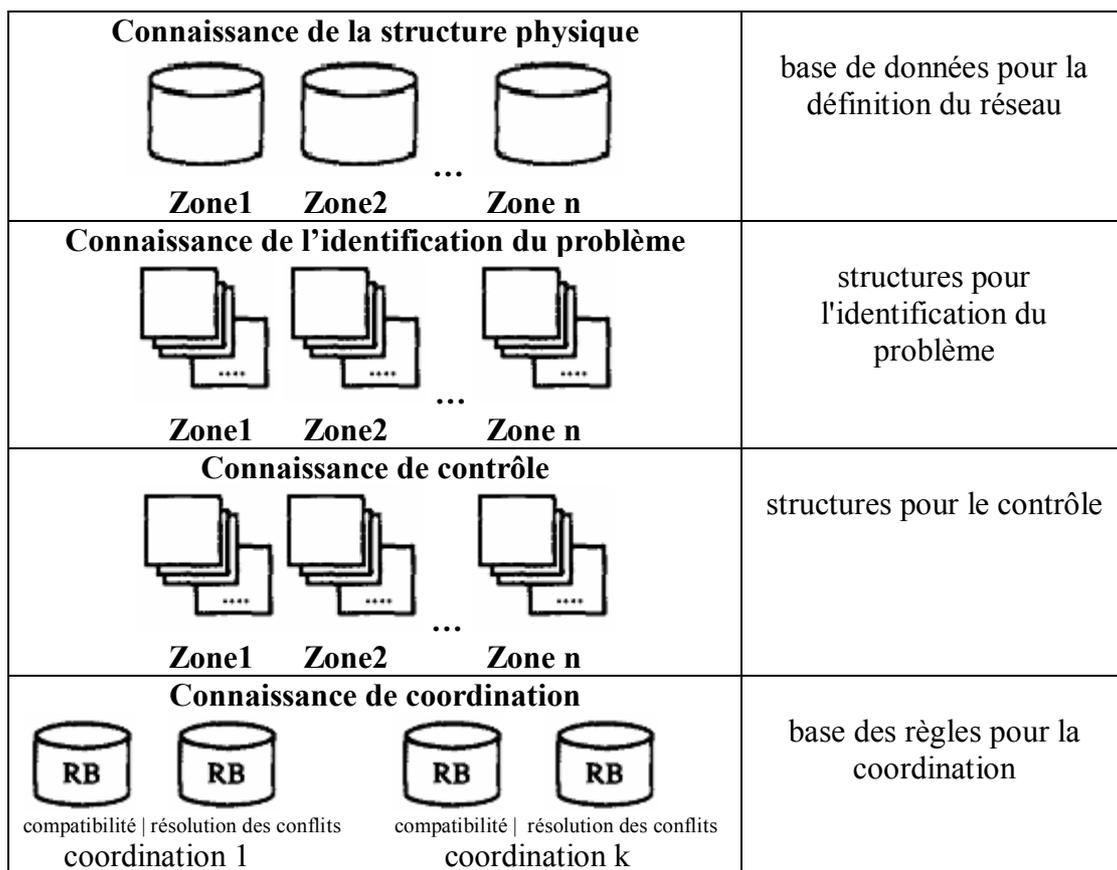


Figure 2-1 : une architecture générale de TRYS.

3-1-3- KSM

KSM (Knowledge Structure Management) est un environnement logiciel qui est basé sur une méthodologie pour la construction et la réutilisation de modèles de connaissance. La

méthodologie est outil utile pour ceux qui doivent établir de grands et complexes modèles de connaissance dans des projets réels.

Selon (Cuenca *et al*, 2000), il y a trois principes d'organisation pour la structure d'une application basée sur la connaissance :

1. Principe orienté-tâche : une tâche est définie comme un but à réaliser, elle est décrite par un type d'entrée qu'elle obtient, et un type de sortie qu'elle fournit; la tâche est décomposée en sous-tâches donnant un arbre qui montre la structure générale du modèle. Une tâche est associée à un ensemble de méthodes par lesquelles un but est réalisé.
2. Principe orienté-domaine : c'est l'utilisation d'une ontologie pour décrire les spécifications des éléments du domaine (dans une base de données), ce qui peut faciliter le partage des connaissances.
3. Combiné : zone de connaissance similaire au principe de la programmation orienté objet : des entités logicielles contenant des données et des processus.

Quatre catégories de questions sont considérées dans ce modèle:

- Classification : Qu'est-ce qui se passe ? (connaissance du problème)
- Diagnostic : Pourquoi cela s'est passé ? (connaissance du comportement)
- Prédiction : Que se passera-il si... ? (connaissance de régulation)
- Configuration : Quoi faire si ... ? (connaissance de régulation)

Le concept de base dans l'environnement KSM est la "zone de connaissance", ce concept est composé de deux parties : **une connaissance** représente l'ensemble de la connaissance des zones et **un fonctionnement** représente l'ensemble des tâches avec leurs méthodes.

KSM utilise une organisation combinée "Tâche-méthode-domaine". Le modèle du domaine peut être formulé par : 1) un vocabulaire conceptuel, 2) des relations entre ces concepts. KSM a proposé deux langages pour formuler les caractéristiques du modèle :

3-1-3-1- Le langage *Concel* :

Il est utilisé pour formuler le vocabulaire conceptuel du domaine étudié (des concepts ayant des attributs et une marge pour les valeurs possible de ces attributs)

Exemple : pour exprimer la capacité d'une voie dans le réseau de trafic urbain :
Capacity (INTERVAL RANGE 0-2000) [vehicle_Km]

Le tableau 2-4- montre le vocabulaire de base pour le modèle du domaine de trafic avec le langage *Concel* :

Classes de concepts	Attributs	Valeur
Détecteur	Concentration Vitesse Circulation	(Valeur en %) (Valeur en km/h) (Valeur en véh/h)
Intersection	Capacité niveau de Saturation régime de circulation Détecteur	Numéro en véh./h {libre, critique} {fluid, unstable, congested} identifiant de détecteur
Lien	Section d'entrée Section de sortie	Instance de section Instance de section
Trajet	liens temps de voyage	liste d'instance de liens (minutes)
PMV	Message	{“Bouchon à MN km”, “pour arriver à la zone A, 20 mm par 137, }

Tableau 2-1 : le vocabulaire utilisé dans le modèle de trafic dans KSM.

3-1-3-2- Le langage Link

Il est utilisé pour formuler des méthodes de résolution des problèmes. Une méthode correspond à une stratégie de résolution du problème, elle est divisée en deux niveaux :

- a) Flux de données (entrées et sorties) : connexion de données entre les sous-tâches.
- b) Flux de contrôle (paramètres et états) : l'ordre de l'exécution des sous-tâches, les paramètres déterminent comment une tâche doit fonctionner.

Le tableau 2-5 montre le format général d'une méthode avec le langage *Link*, une méthode est composée des parties suivantes :

METHOD method-name ARGUMENTS [Input list-of-inputs] [Output list-of-outputs] DATAFLOW data-connection-among-sub-tasks [CONTROL TASKS data-connection-among-control-tasks] CONTROL FLOW rules-to-determine-the-control-regime [PARAMETERS default-values-for-parameters]
--

Tableau 2-2 : un format général d'une méthode avec le langage *Link*.

Exemple :

Dans les STU la tâche de *classement* peut prendre en entrée des mesures et elle donne en sortie une catégorie d'état par rapport aux degrés de similarité. Le degré de similarité peut être considéré comme un paramètre de contrôle.

En résumé, l'environnement KSM a proposé deux langages, *Concel* pour exprimer les concepts du domaine étudié et *Link* pour exprimer les méthodes de résolution de problèmes. Les deux langages utilisent des notions et des principes similaires à l'approche orientée-objet pour faire la modélisation.

3-3- InTRYS et TRYS A2 (Hernandéz et al, 2002)

(Hernandéz et al, 2002) ont proposé deux architectures Multi-agents pour un système intelligent de pilotage de trafic basé sur l'architecture TRYS (TRYS est un environnement basé sur les agents qui sert à construire des systèmes intelligents de pilotage de trafic (Cuenca et al, 1998), voir 2-2)

- 1- InTRYS (Intégré) : il réalise la coordination entre les agents en utilisant un mécanisme centralisé; il a été implémenté dans la ville de Barcelone en Espagne.
- 2- TRYS_{A2} (Agent Autonome) : la coordination est faite par des interactions entre des agents autonomes; cette extension a été utilisée uniquement dans des laboratoires pour des objectifs de recherche.

Ces deux systèmes sont des systèmes d'aide à la décision pour une gestion de trafic temps réel, ils utilisent les mêmes techniques de raisonnement depuis leur base de connaissance.

Ils fonctionnent sur trois niveaux d'opération : capteurs, données, connaissances.

3-3-1- Les agents de pilotage de trafic

L'objectif de ces agents est de fournir deux types d'informations :

- 1) des diagnostics et des explications des problèmes;
- 2) des propositions d'actions de contrôle vers les feux ou vers les panneaux.

L'agent de pilotage de trafic possède quatre types de connaissance :

- 1) Des scénarios de problèmes possibles
- 2) Des propositions de contrôle de trafic
- 3) La structure du réseau
- 4) L'histoire de la demande du trafic

Le modèle de connaissance est une collection d'unités de connaissance (UC), Trois types d'UC sont utilisés :

- L'UC du comportement : elle est composée de deux sous-unités :
 1. Une structure physique : des capteurs – détecteurs à partir desquels on collecte les variables globales du système comme la vitesse, la concentration et l'écoulement.
 2. Une connaissance de la demande : des informations historiques concernant la période, le type de la journée, l'intervalle et le nombre estimé des usagers depuis une origine vers une destination
- L'UC des problèmes de trafic : chargés de détecter et expliquer les différentes situations rencontrées. Les objectifs de cette UC sont :
 - 1) d'analyser les données capturées pour signaler un problème
 - 2) d'expliquer si le problème est grave et quelles sont les causes.

Le déséquilibre entre la demande et la capacité du réseau augmente la concentration dans certaines sections du réseau. Par conséquent, la vitesse d'écoulement va se réduire.

Les informations nécessaires pour un problème détecté sont :

- 1- Où est le problème ? des travaux, une entrée à une grande route qui a une demande importante, tourner à gauche (TAG) avec durée de vert non suffisant, un accident, ...
 - 2- Quelle est la gravité du problème (veh/h)
 - 3- Quelle est la cause ?
- L'UC du contrôle de trafic : dont l'objectif est de proposer des messages à appliquer sur les panneaux à messages variables, et de modifier les plans de feux pour diminuer la demande au point posant le problème.

3-3-2- Modèles de coordination

Les agents de pilotage de trafic dans les deux systèmes InTRYs (Intégré) et TRYSA₂ ont la même structure et ils appliquent les mêmes techniques, mais la coordination entre agents n'est pas la même : la structure de InTRYs est une structure centralisée où l'agent coordinateur réalise la coordination entre les agents. Dans TRYSA₂ (Agent Autonome) : la coordination est faite par des interactions entre les agents (structure décentralisée).

La figure 2-2 montre la différence entre les deux :

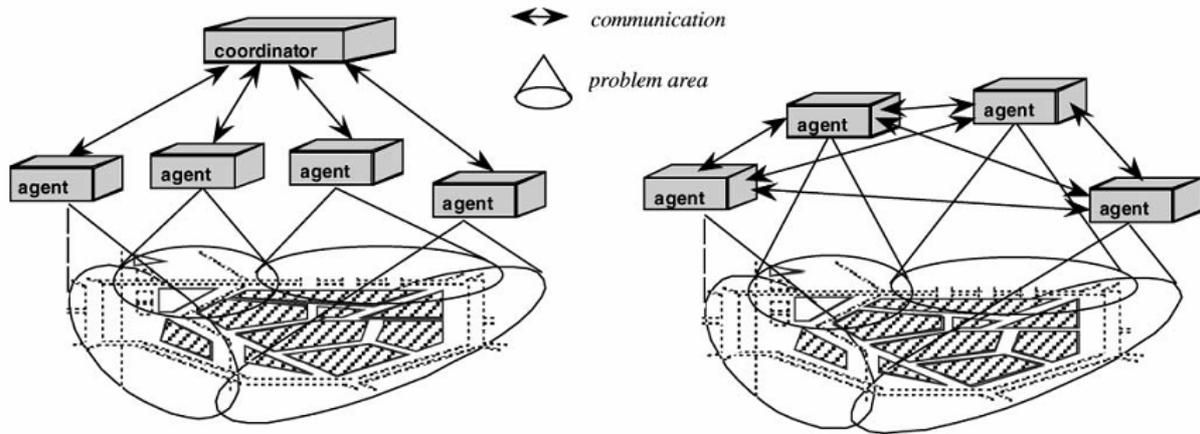


Figure 2-2 : InTRYS (Centralisé) et TRYS A2 (Décentralisé)

1. InTRYS (une coordination centralisée) :

Les agents envoient leurs propositions de changement de phasage au coordinateur qui cherche les conflits entre ces propositions et suggère une solution cohérente pour l'ensemble du réseau.

2. TRYS A2 (une coordination décentralisée) :

Les agents dans cette architecture sont autonomes. Ils essaient d'améliorer la situation du trafic chacun dans sa zone de problème. La solution finale est influencée par les informations échangées entre les agents, si un conflit existe entre la solution finale et la solution locale d'un agent, il reste neutre.

3-4- (Ossowski et al, 2005)

(Ossowski et al, 2005) ont proposé un système d'aide à la décision pour la gestion du trafic basé sur des abstractions multi-agents communicantes et organisées (Modèle de pilotage basé sur les systèmes multi-agents). La figure 2-3 montre l'architecture abstraite du système proposé.

Afin de profiter des informations disponibles sur le réseau de trafic urbain, l'utilisation d'un système multi-agents est proposée où les centres de décision sont représentés par des agents, ce qui réduit la complexité de la conception et permet une solution basée sur les dialogues entre les différentes entités et avec le décideur.

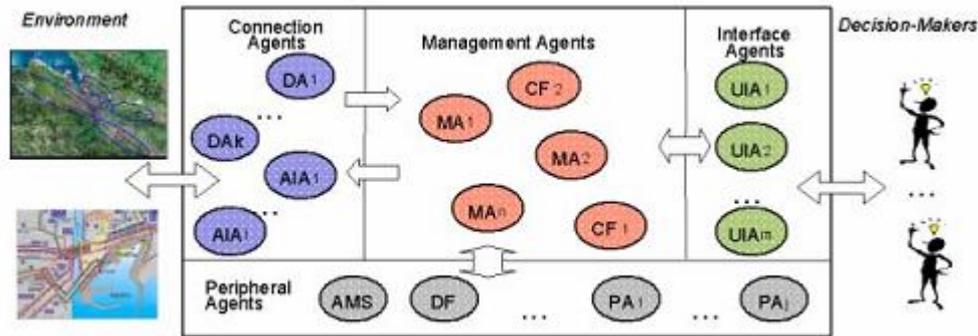


Figure 2-3 : une architecture abstraite d'un système d'aide à la décision (Ossowski *et al*, 2005).

Le modèle d'organisation :

Dans l'architecture proposée on peut trouver les types d'agents suivants :

- L'agent de données DA, dont le rôle est de fournir des informations sur l'état actuel du système étudié,
- L'agent d'implantation des actions AIA, responsables de l'exécution des décisions prises par le décideur,
- L'agent de management MA; qui joue le rôle d'informer, de conseiller, d'expliquer ...,
- L'agent d'interface utilisateur UIA, qui est une couche intermédiaire entre le système d'aide à la décision et le décideur,
- L'agent de périphérique PA, qui représente l'infrastructure physique du système étudié,

D'autres agents peuvent être trouvés dans cette organisation : l'agent de détection des problèmes PDA pour montrer le trafic et détecter les cas de saturation, l'agent de contrôle CA pour générer des actions possibles pour traiter la situation, l'agent de connexion et l'agent de gestion.

Cette architecture est basée sur un dialogue entre le système d'aide à la décision et le décideur, les dialogues permettent au système de fournir à l'utilisateur des informations, des explications, des propositions et des conseils.

Cette architecture a été appliquée pour détecter les accidents et pour proposer des messages aux usagers sur ces accidents. Le modèle est utilisé à Bilbao en Espagne, les données traitées par cette architecture sont obtenues par le simulateur AIMSUN (Algers *et al*, 1997).

Un autre exemple a été traité pour gérer le fonctionnement des bus de la ville de Malaga.

Remarques :

Dans l'étude proposée (Ossowski *et al*, 2005) il n'y a pas d'agents qui présentent les centres de décision. On y trouve beaucoup de types d'agent. Cette étude ne propose pas une méthodologie de modélisation prenant en compte les aspects physique et logique des STU. Deux modèles ont été proposés, chacun pour traiter un seul type de problèmes; le premier est dédié à la congestion et aux panneaux aux messages variables, et le deuxième traite les problèmes des bus. Les modèles proposés utilisent un simulateur externe « AIMSUN2 ».

3-5- CMTMRGS (Adler et Blue, 2002)

(Adler et Blue, 2002) ont proposé une approche multi-agents en temps réel de coopération entre le système de gestion et le système de guidage. Cette approche est basée sur un principe de négociation entre les agents représentant des entités de régulation, des fournisseurs d'information et les conducteurs.

Cette étude suppose que les véhicules sont équipés d'un système intelligent d'information responsable du guidage et du conseil. Un centre de service d'information (ISP) collecte, traite et stocke les informations, donne aux usagers des conseils et des informations sur les conditions temps réel de trafic, et génère des plans de trajets à partir des requêtes des usagers concernant leur origine et leur destination.

Concernant la qualité de service dans le domaine du trafic, les usagers et les gestionnaires ont des objectifs différents à satisfaire, donc les solutions suggérées doivent être acceptables des deux côtés.

CMTMRGS:

CMTMRGS (Cooperative Multi-agent Transportation Management and Route Guidance System) (Adler et Blue, 2002) est un système multi-agents coopératif de guidage et de gestion du trafic, basé sur trois couches (3 niveaux) :

- 1) Un centre de gestion de trafic TMC
- 2) Un fournisseur de services d'information ISP, couche intermédiaire entre le TMC et l'ITIS (*Intelligent Traveler Information System*) dont l'objectif est de garder les informations personnelles des voyageurs et de réaliser les communications avec les décideurs et le TMC.
- 3) Un système intelligent d'informations pour le voyageur ITIS responsable des tâches suivantes :
 - a. Apprendre les préférences du voyageur
 - b. Collecter les informations de trafic

- c. Trouver le meilleur trajet pour une destination donnée
- d. Négocier les meilleurs trajets avec les ISP
- e. Faire le guidage

Cette architecture est un essai de traitement des voyageurs et des décideurs comme des entités coopératives. Chaque usager déclare son origine, sa destination, son trajet préféré, sa vitesse, l'heure de départ et les contraintes de temps s'il y en a. Les agents de contrôle calculent un trajet possible pour les véhicules à partir des conditions de trafic pré-testées.

Tous les carrefours à feux négocient entre eux et avec les voitures qui approchent, selon la longueur et l'instant de départ des cycles de feux afin de réduire au minimum le temps d'attente pour les voitures.

Selon les résultats de (Adler et al, 2002), les optimisations locales à chaque carrefour n'amèneront pas à des résultats globalement optimaux. L'influence des carrefours entre eux doit être prise en compte.

Les défauts de cette approche :

- 1- La coopération entre les usagers et les ISP est difficile.
- 2- Les usagers ne veulent pas partager leurs informations personnelles concernant leurs trajets.
- 3- Les usagers ne veulent pas payer plus d'argent pour de nouveaux services.

3-6- Divers:

(Li *et al*, 1996) ont suggéré un système coopératif de contrôle et de régulation basé sur les SMA, et ont proposé une architecture en trois niveaux hiérarchiques : global, groupe et individuel.

(Finder et Strap, 1992) ont présenté une approche distribuée pour optimiser la signalisation du trafic. (Iftar, 1997) a développé une programme non-linéaire basé sur une approche intelligente de contrôle pour décentraliser la régulation du trafic.

Les outils d'aide à la décision mentionnés précédemment fonctionnent soit avec des données réelles de trafic, soit avec des simulateurs de trafic. Les simulateurs ne sont pas l'intérêt essentiel de ce travail, mais nous en présentons deux, puisque ce travail s'intègre finalement dans un environnement de modélisation dans lequel un modèle de simulation de trafic est déjà réalisé.

4- Simulateurs :

Les simulateurs sont des outils qui aident les experts du domaine à analyser certaines situations et à comprendre le fonctionnement du système. Nous présentons TRANSIMS et AIMSUN2 comme exemple de ce type d'outils, pour plus d'informations sur ces simulateurs et sur d'autres, (Sarramia, 2002) est une meilleure référence.

4-1- TRANSIMS

TRANSIMS "TRansportation ANalysis and SIMulation System" est un ensemble de procédures pour modéliser des déplacements pour satisfaire les besoins du MPO (Metropolitan Planning Organization) et du US DOT (Department of Transportation).

TRANSIMS produit des données détaillées sur le trajet, la congestion, et la pollution; il simule les déplacements et les comportements des individus, et il a la possibilité d'évaluer des scénarios complexes.

Il est basé sur quatre modules :

- Un estimateur de population,
- Un générateur d'activité,
- Un planificateur d'itinéraire,
- Un microsimulateur du trafic.

En utilisant ces composants, TRANSIMS estime les activités des individus et des déplacements, il propose des plans de déplacements satisfaisant ces activités, et il associe des parcours pour chaque origine - destination, et finalement réalise une micro-simulation de tous les déplacements des individus, et des éléments du système de transport.

Les concepts et structure de TRANSIMS incluent une représentation cohérente et continue du temps; et une représentation détaillée des déplacements des personnes. Les modèles de TRANSIMS ont été appliqués sur des données de Dallas, au Texas, et de Portland, en Orégon.

TRANSIMS a été développé par des chercheurs du laboratoire national de Los Alamos et il est commercialisé par le centre de solution de 'IBM TRANSIMS'.

4-2- AIMSUN2

AIMSUN2 "Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks" est un outil qui peut reproduire les vraies conditions du trafic sur un ordinateur. Il est principalement utilisé pour examiner de nouveaux systèmes de commande du trafic ou pour évaluer différentes options pour mettre en application une nouvelle infrastructure avant de la construire. AIMSUN2 suit une approche microscopique de simulation. Ceci signifie que

le comportement de chaque véhicule simple dans le réseau est modélisé tout au long de la durée de simulation (<http://www.transver.de>).

4-2-1- GETRAM

GETRAM “Generic Environment for Traffic Analysis and Modeling” est un environnement de simulation. Le micro-simulateur du trafic AIMSUN2 fait partie de cet environnement.

Il est possible de lier GETRAM avec EMME/2 pour faire des analyses macroscopique et microscopique en utilisant les mêmes données du trafic (comme les matrices origine-destination)

4-2-2- EMME/2

EMME/2 sert à planifier les déplacements de personnes aux échelles urbaine et régionale. Le progiciel EMME/2 sert pour la planification du transport des personnes sur des réseaux multi-modaux. Il offre aux planificateurs un ensemble complet et flexible d'outils pour la modélisation et l'analyse de la demande, ainsi que pour l'analyse et l'évaluation des réseaux.

Tous les modes de transport sont intégrés en un réseau cohérent, et l'interaction est prévue entre les véhicules particuliers et de transport en commun.

EMME/2 propose des outils de manipulation de matrices Origine - Destination, qui permettent la mise en application de nombreux modèles de prévision de la demande; des procédures d'affectation; des calculs interactifs, qui permettent la mise en œuvre de méthodes d'évaluation et d'analyses d'impact; le langage utilisé permet d'automatiser des opérations répétitives; des outils graphiques d'édition de réseau.

EMME2 est basé sur une approche macroscopique.

5- Cadre d'évaluation

Dans cette section, nous définissons un cadre d'évaluation des outils et des approches existantes dans la littérature qui concernent le pilotage ou les modèles de pilotage des STU. Pour chaque outil étudié nous décrivons :

- 1) Le problème pris en compte.
- 2) L'objectif et l'approche utilisée.
- 3) Une évaluation des résultats.

5-1- Le problème pris en compte

Cette description comprend les activités couvertes par l'étude concernée, et l'horizon temporel de l'application (jusqu'à quelle étape dans le processus d'aide à la décision l'application est utilisée ?):

a. **Les activités de régulation** que l'outil prend en considération : nous rappelons qu'une entité de régulation dans les STU gère quatre activités (applications) (voir chapitre 1 : 2-2) : carrefour à feux (CAF), affichage de messages (PMV), détection de véhicule de transport en commun (VTC), mesure du trafic (MDT). Le tableau 2-1 montre quelles activités sont couvertes par les outils et les approches trouvées dans la littérature. Pour comparer les différents outils, nous utilisons un critère simple qui consiste à compter le nombre de liens opérationnels entre les applications couvertes (Le nombre maximum étant 6) :

	CAF	PMV	VTC	MDT	Intégralité
TRYS [Cuenca et al. 2000]	X			X	1/6
[InTRYS, 2002]	X	X		X	3/6
[TRYS A2, 2002]	X	X		X	3/6
[Adler et al, 2005]	X			X	1/6
[Ossowski et al, 2005]	X	X	X	X	4/6
[Li et al, 1996]	X				
[Finder et Strap, 1992]	X				
[Iftar, 1997]	X				
[Van Katwijk, 2004]	X	X			1/6
[TRANSYT, 2005]	X			X	1/6
[CLAIRE, 2002]	X	X		X	3/6
[SCOOT, 2005]	X			X	1/6

CAF : Carrefour A Feux, MDT Mesure De Trafic,

PMV : Panneaux à Messages Variable, VTC : Véhicule de Transport en Commun.

Tableau 2-3 : Les activités de régulations traitées

b. **L'horizon temporel** i.e., le niveau de planification considéré par rapport aux objectifs envisagés : **opérationnel, tactique, stratégique**.

c. **La granularité** : Les informations sur les entités peuvent être détaillées (dites **microscopiques**) ou peu détaillées et dites **macroscopiques**. Une catégorie intermédiaire dite **mesoscopiques** traitent les données par paquets et non individuellement.

Le tableau 2-2 montre la position des applications par rapport à l'horizon temporel et la granularité.

	Macroscopique	Mesoscopique	Microscopique
Stratégique	EMME/2		Transims Getram
Tactique		[TRYSA2, 2002] [Adler et Blue, 2002]	[Ossowski, 2005] [InTRYS, 2002] CLAIRE
Opérationnel		[TRYSA2, 2002] [Li et al, 1996]	[Ossowski, 2005] [Iftar, 1997] [Findler et Strap, 1992] TRANSYT SCOOT

Tableau 2-4 : La position des applications par rapport à l'horizon temporel et spatial (la granularité).

5-2- L'approche utilisée pour résoudre chaque problème

Les outils sont ici classés en fonction de leur objectif, des approches utilisées pour leur conception et enfin de la qualité des résultats qu'ils fournissent.

a. L'utilisation envisagée (objectifs) :

Les outils étudiés sont utilisés comme un **système d'aide à l'exploitation** ou comme un **système d'aide à la décision**.

b. Approche:

La plupart des applications étudiées sont basées sur une **approche multi-agents**. Dans ce cas nous différencions les structures statiques et dynamiques du SMA.

Nous indiquons quel type de **coordination** entre les agents est utilisé par trois attributs :

- i. Le contrôle : centralisé ou distribué (décentralisé)
- ii. Structure : dynamique ou statique
- iii. Attitude (comportement) : coopérative, compétitive ou les deux

Il existe d'autres applications basées essentiellement sur l'approche orientée-objet (OOP).

5-3- Evaluation des résultats

a. La maturité (niveau d'expérience) :

Nous considérons les quatre niveaux d'expérience suivants :

- i. Proposition architecturale : une description de l'idée et des principes avec leurs caractéristiques.
- ii. Simulation : l'outil a été validé dans un environnement de simulation avec de données réelles ou simulées.
- iii. Environnement réel : l'outil a été validé dans un environnement réel avec de données limitées ou complètes.
- iv. Déploiement : c'est le niveau le plus élevé où l'application a été implémentée dans le monde réel et a été utilisée.

b. L'évaluation des résultats :

- i. Qualitative (entre les propriétés des applications étudiées)
- ii. Quantitative (de types différents d'expérience).

	Utilisation (Objectif)	Type	coordination	Maturité	Type de données	Type de résultats
TRYS [Cuena et al. 2000]	SAD	SMA dynamique	Distribuée, Coopérative	Simulation	Artificielles, limitées	0
[InTRYS, 2002]	SAD	SMA statique	Centralisée, Coopérative	Déploiement	Réelles, limitées	2
[TRYS A2, 2002]	SAD	SMA statique	Distribuée, Coopérative	Simulation	Réelles, complètes	2
[Adler et al, 2005]	SAD, SAE	SMA dynamique	Distribuée, Coopérative	Simulation	Artificielles, complètes	Qualité
[Ossowski et al, 2005]	SAD	SMA	Distribuée, Coopérative	Simulation	Artificielles, limitées	0
[Li et al, 1996]	SAD	SMA statique	Distribuée, Coopérative	Simulation	Artificielles, limitées	0
[Finder et Strap, 1992]	SAD	Contrôle distribué	Distribuée	Proposition architecturale	-	-
[Iftar, 1997]	SAD	LP		Proposition architecturale	-	-
[van Katwijk, 2004]	SAE	SMA	Distribuée, Coopérative	Simulation	Artificielles, limitées	Qualité
[TRANSYT, 2005]	SAE différé	OOP		Déploiement	Réelles, limitées	-
[CLAIRE, 2002]	SAE réel	OOP (C++)		Déploiement	Réelles, limitées	-
[SCOOT, 2005]	SAE réel	OOP		Déploiement	Réelles, limitées	-

SAE : Système temps réel d'aide à l'exploitation, SAD : système d'aide à la décision

Tableau 2-5 : L'approche utilisée et l'évaluation de résultats.

6- Conclusion

Dans les années 90 un système d'aide à la décision faisait essentiellement de la gestion de données, mais ce qui importe à présent c'est plutôt la signification de ces données. Un système d'aide à la décision pour un décideur correspond à une base de connaissance du système.

Le STU est distribué physiquement et logiquement, ce qui impose l'utilisation d'une approche SMA distribuée, où les agents sont semi-autonomes, ils récoltent et fournissent des données qui aident à la décision.

L'approche multi-agents semble très répandue dans ce domaine. Mais les modèles de pilotage qui existent dans la littérature manquent d'aspect méthodologique et d'un environnement de modélisation. Pour cette raison, il n'y a pas un outil qui prend en compte l'ensemble des activités des entités de régulation, leur intégration dans un modèle de domaine comprenant les aspects physiques et logiques. En plus, chaque modèle proposé est dédié à un seul type de problème, ce qui donne une vision incomplète de l'ensemble.

On peut également constater que la majorité des applications qui sont basées sur les SMA sont orientées-véhicule, autrement dit, les agents représentent des véhicules. Cette modélisation nous semble incomplète puisque le comportement global des STU est principalement influencé par les règles de fonctionnement qui ne sont pas modélisées par les algorithmes de résolution des agents.

Dans le chapitre suivant nous présentons les notions d'agent, des systèmes multi-agents, et quelques outils, méthodes et plateformes dédiés à cette approche.

Chapitre 3

- L'approche multi-agents

Sommaire

INTRODUCTION	55
1- CONTEXTE	55
2- CONCEPTS.....	56
2-1- <i>Agent vs Objet</i>	58
2-2- <i>Agents vs système expert</i>	58
3- PROPRIÉTÉS ET TYPES D'AGENTS	59
3-1- <i>Agent réactif</i>	59
3-2- <i>Agent cognitif</i>	59
3-3- <i>Agent hybride</i>	60
4- MÉTHODOLOGIES BASÉES SUR LES SMA	60
4-1- <i>Gaia</i>	60
4-2- <i>Tropos</i>	62
4-3- <i>MaSE</i>	63
4-4- <i>Discussion et comparaisons des méthodologies</i>	68
5- OUTILS ET PLATEFORMES POUR LES SMA.....	69
5-1- <i>L'approche AUML</i>	69
5-2- <i>FIPA et la plateforme JADE</i>	70
5-3- <i>La plateforme MadKit</i>	71
6- INTERACTIONS, COMMUNICATIONS ET COORDINATION.....	76
7- CONCLUSION.....	77

Introduction

L'intelligence artificielle distribuée et l'approche multi-agents sont très répandues pour l'étude des systèmes complexes.

Le but de ce chapitre est de rappeler la notion d'agent et de système multi-agents, et de présenter quelques méthodologies basées sur cette approche, ainsi que des plateformes et des outils pour construire les systèmes multi-agents.

1- Contexte

L'intelligence artificielle distribuée (IAD) est une approche possible pour contrôler les systèmes complexes en décomposant et en distribuant la prise de décision. Les systèmes complexes sont décomposés en sous-ensembles en relation. Chaque sous-ensemble est responsable du contrôle de son domaine et de la coordination des activités avec les autres sous-ensembles (Adler et Blue, 2002).

Une approche basée sur les agents permet de concevoir des modèles de pilotage non centralisés et auto organisés permettant d'améliorer la réactivité des systèmes de décision ce qui accroît leur autonomie. Le terme "agent" dénote un matériel ou un logiciel qui soutient un contexte de prise de décision autonome. Les agents ne sont pas simplement des objets ou des acteurs, mais également des entités autonomes essayant d'atteindre des buts en agissant sur leur environnement. Selon (Wooldridge M., Jennings N.R., 1995), des agents doivent avoir les propriétés suivantes : autonomie, proactivité, adaptabilité, sociabilité et mobilité.

D'après (Burmeister et al, 1997), les Systèmes Multi-Agents (SMA) sont des outils pour conceptualiser et décrire les systèmes complexes. Cette approche va contribuer à la réduction de la complexité de la modélisation classique. Trois conditions pour lesquelles les technologies d'agents peuvent aider dans la conception et l'analyse sont à prendre en compte :

- 1- Le domaine du problème est géographiquement distribué,
- 2- Des sous-ensembles existent dans un environnement dynamique,
- 3- Les sous-ensembles doivent interagir entre eux plus efficacement.

Les technologies de l'intelligence artificielle distribuée (IAD) et les approches de résolution coopératives des problèmes peuvent aider à traiter les problèmes de ce type

de domaine. Elles offrent certains avantages pour la résolution des problèmes : une réponse plus rapide, une flexibilité accrue, une robustesse, le partage de ressources, et une meilleure adaptabilité (Ferber, 1999, 2004), (Drogoul, 1998, 2002).

2- Concepts

Il n'y a pas une définition universelle pour un agent, mais un consensus général admet que l'autonomie est essentielle pour l'agent, d'autres caractéristiques varient d'un système à l'autre comme l'apprentissage, la sociabilité, ...etc.

(Wooldridge et Jennings, 1995) ont défini l'agent comme suit : "Un agent est un système informatique situé dans un environnement quelconque capable d'exécuter des actions autonomes dans son environnement pour atteindre ses objectifs". Une autre définition a été suggérée par (Huhns et Singh, 1998) : "Un agent est une entité logicielle qui est capable de percevoir, raisonner, réagir et communiquer."

L'agent ne peut pas complètement contrôler son environnement car il a une vision limitée de cet environnement, donc il ne peut le contrôler que partiellement. La figure 3-1 montre l'architecture d'un agent qui est composée de deux parties : une partie de perception et une partie d'action, la figure montre aussi les relations entre l'agent et son environnement :

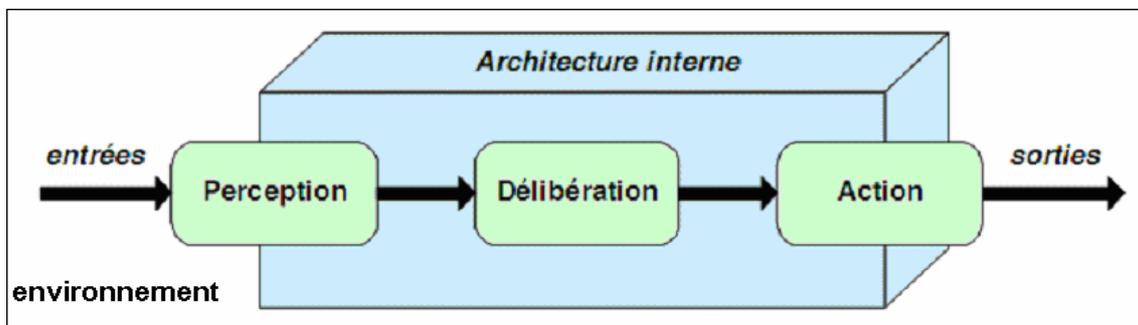


Figure 3-1 : un agent (perception et action sous-systèmes) dans son environnement (Ferber, 1999).

Un **agent** est défini par un ensemble de perceptions (entrées), un ensemble d'actions (sorties) et une architecture interne. Les termes *senseurs* et *effecteurs* sont utilisés pour désigner les moyens par lesquels un agent interagit avec son environnement.

La définition la plus simple d'un **système multi-agents** est la suivante : un SMA est un ensemble d'agents partageant un environnement commun. Un environnement

définit les conditions d'existence de chaque entité, et il met en relation l'ensemble des agents.

Selon (Demazeau, 1995), (Demazeau, 2001) un SMA est un ensemble d'agents potentiellement organisés qui interagissent dans un environnement commun. (Il a ajouté les notions d'organisation et d'interaction).

L'approche VOYELLES (Demazeau, 2001) consiste à considérer que l'étude des SMA peut se faire en fonction de quatre aspects fondamentaux :

- L'agent : architectures internes des agents
- L'environnement : le milieu dans lequel évoluent les agents
- L'interaction : les moyens par lesquels les agents interagissent
- L'organisation : les moyens utilisés pour structurer l'ensemble des entités

Chacun de ces aspects définit une problématique qui doit être étudiée dans chaque étape de l'étude, de la phase d'analyse à l'implémentation.

Les **environnements** peuvent être classifiés selon plusieurs types et propriétés (Russell et Norvig, 1995) :

- Un environnement peut être **accessible** si l'agent a des informations complètes à propos de son état, mais dans la majorité des environnements du monde réel, les agents possèdent une connaissance partielle de l'état de l'environnement.
- Un environnement est **déterminé** si une action a un effet unique et précis.
- Un environnement **statique** est un environnement dont l'état ne change pas au fil du temps. Le monde physique est un exemple d'environnement **dynamique** où plusieurs processus fonctionnent et modifient son état en dehors du control des agents.
- Un environnement est **discret** s'il contient un nombre fixe et fini d'actions et de perceptions, sinon il est **continu**.

Généralement, les environnements les plus complexes sont ceux qui sont inaccessibles, non-déterminés, dynamiques et continus. Nous pouvons les nommer comme des *environnements ouverts*. Par exemple, l'environnement des systèmes du trafic urbain est un environnement ouvert ayant les quatre caractéristiques.

La complexité de la conception d'un système multi-agents correspond à la complexité de son environnement et à la nature des interactions entre les agents et leur environnement.

2-1- Agent vs Objet:

Les programmeurs familiers avec la programmation orientée objet comme C++ ou Java ne voient pas de nouveauté concernant l'approche agent, cependant, il est difficile d'imaginer un objet capable de décider ce qu'il veut ou demander des informations à d'autres objets.

La programmation orientée objet se distingue par le concept de l'héritage, qui permet à de nouvelles classes d'être développées à partir des classes existantes simplement en modifiant les opérations et les attribues. Une classe peut hériter des opérations et du comportement d'une classe de base.

Les superclasses fournissent la fonctionnalité qui est commune à toutes ses sous-classes tandis qu'une sous-classe fournit la fonctionnalité additionnelle pour spécialiser son comportement.

La distinction entre un agent et un objet est faite selon trois caractéristiques (Wooldridge, 2002) :

- un agent encapsule une notion d'autonomie plus forte que celle d'un objet. Un agent peut décider lui-même s'il va exécuter une action requise par d'autres agents ou non.
- Les agents possèdent un comportement flexible (réactif, proactif, social). Un objet standard n'a pas ce type de comportement.
- Un système multi-agents est un système *multi-thread* dans lequel chaque agent a au moins un *thread* de contrôle.

2-2- Agents vs système expert

Considérant les précédentes caractéristiques des agents, un système multi-agents semble très similaire à un système expert, d'où l'importance de les distinguer :

Les systèmes experts sont la technologie la plus importante de l'Intelligence Artificiel des années 80s. Un système expert est un système capable de résoudre des problèmes ou de donner des conseils selon sa connaissance du domaine.

- Le système expert n'appartient pas à un environnement, il obtient ses informations par un utilisateur et ne réagit pas sur son environnement.
- Un système expert n'a pas un comportement réactif ou proactif.
- Un système expert n'a pas une capacité sociale dans le sens de coopération, coordination ou négociation.

3- Propriétés et types d'agents

L'intelligence de l'agent:

Il est difficile de définir l'intelligence et comment un agent peut être intelligent. Pour répondre à cette question il est possible de donner une liste des propriétés qu'un agent doit avoir, (Wooldridge et Jennings, 1995) ont suggéré la liste suivante :

- Réactivité, un agent intelligent est capable de percevoir son environnement et de réagir aux changements pour réaliser ses objectifs.
- Proactivité, les agents intelligents ont un comportement autonome, capable de prendre des initiatives et orienté vers leurs propres buts.
- Sociabilité, les agents intelligents interagissent entre eux, avec leur environnement (et peut-être avec le décideur) pour coopérer, coordonner et négocier.

D'autres suggestions

- Autonomie : l'agent décide lui-même ce qu'il va faire pour atteindre ses objectifs.

A partir de ces caractéristiques on peut identifier trois types d'agents : réactif, cognitif et hybride :

3-1- Agent réactif

Certains types d'agents décident quoi faire sans apprendre de leur histoire, ils réagissent toujours en fonction de leur état interne et de l'événement reçu de l'environnement. De tels agents sont *des agents réactifs* purs.

3-2- Agent cognitif

Les agents cognitifs sont capables de mémoriser leur passé pour appréhender leur avenir et pour mieux choisir les actions qu'ils doivent effectuer. De plus, ils réagissent par rapport à leurs propres buts. Ces agents sont *des agents cognitifs*.

3-3- Agent hybride

La difficulté pour construire un système multi-agents est de garantir une balance pour chaque agent entre la réalisation de ses propres buts (un comportement autonome orienté -but) et la bonne réaction aux changements de son environnement (un comportement réactif). Par conséquent, le fait de donner à un agent une capacité de réactivité et une capacité de proactivité amène à la notion d'*agent hybride*.

Normalement, l'agent possède un répertoire d'actions possibles, et son problème revient à bien choisir l'action convenable pour réaliser ses objectifs.

4- Méthodologies basées sur les SMA

Dans cette partie nous présentons quelques méthodologies génériques basées sur l'approche agent qui peuvent être utilisées pour étudier les systèmes complexes et les systèmes distribués.

4-1- Gaia

Gaia (Wooldridge et al., 2000) était la première méthodologie complète proposée pour l'analyse et la conception des SMA, la version d'origine de Gaia était plutôt pour les systèmes fermés, et elle a adopté des techniques de notation non-standard. Plusieurs extensions de cette méthodologie ont été proposées pour surmonter les limitations de la version d'origine.

Gaia contient les deux phases d'analyse et de conception mais elle ne contient pas les phases de spécification et d'implémentation.

Le processus de Gaia consiste en la construction de plusieurs modèles : rôle, interaction, agent, service et association. L'objectif de ces modèles est de décrire les deux aspects macro (sociétal) et micro (inter agents) du SMA comme le montre la figure 3-2.

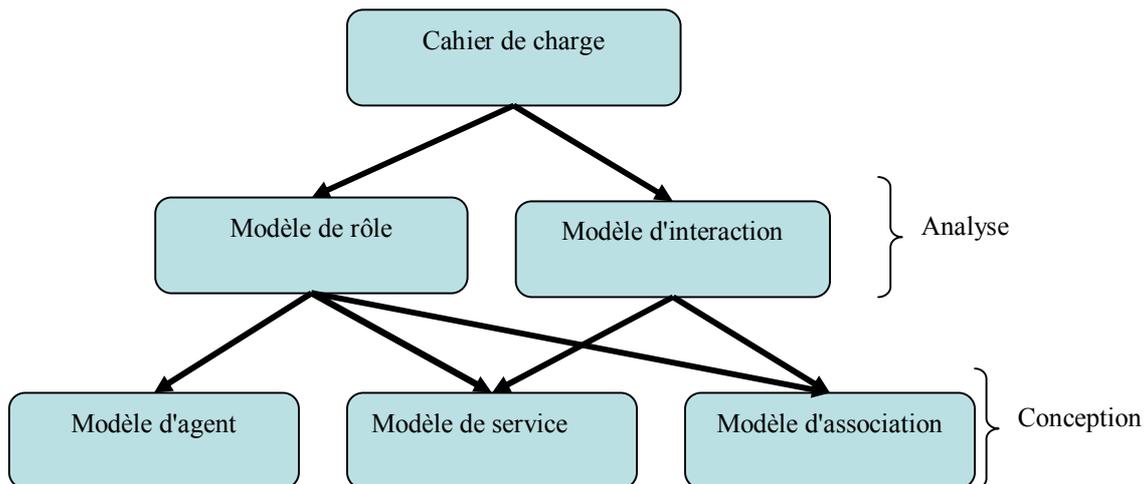


Figure 3-2 : La méthodologie Gaia

L'analyse :

La phase d'analyse de Gaia permet d'identifier les rôles du système étudié et de modéliser leurs interactions. Pour chaque rôle elle décrit le comportement de l'agent dans un schéma semi-formel, l'ensemble de ces schémas définit le rôle complet du modèle. Pour chaque rôle, un schéma de rôle est défini en terme de quatre attributs :

- 1) Permissions : exprimant les ressources disponibles pour le rôle.
- 2) Responsabilités : définissant le fonctionnement du rôle
- 3) Activités : représentant les tâches ou les actions du rôle sans interaction avec les autres rôles.
- 4) Protocoles : représentant les tâches ou les actions du rôle exigeant des interactions avec les autres rôles.

La conception :

Dans cette phase, trois types de modèles sont à produire: le modèle d'agents spécifiant les types d'agents, le modèle de service spécifiant les services à implémenter par les types d'agents et le modèle d'association décrivant les liens de communication entre les agents. Pour construire le modèle d'agent, les rôles seront associés aux agents. Le modèle de service montre les services que chaque type d'agent fournit, ces services sont dérivés à partir des activités et des protocoles des rôles. Pour chaque service il faut spécifier quatre attributs : entrée, sortie, pré-conditions et post-conditions. Le modèle d'association est un graphe orienté entre les types d'agents.

Limitation:

- Gaia ne peut pas modéliser des aspects sociaux importants, ni des structures organisationnelles d'agents dans le système. En plus Gaia suppose que tous les agents sont coopératifs. Pour ces raisons, Gaia ne convient pas aux systèmes ouverts.
- Gaia utilise des notations spécifiques pour représenter les rôles et les protocoles. Malgré la simplicité de ces notations, elles ne sont pas suffisantes pour exprimer des problèmes complexes comme les protocoles d'interaction multi-phases.

4-2- Tropos

Tropos considère les acteurs (agents, rôles ou positions), les buts, les dépendances de l'acteur comme des primitives de conception pour modéliser un système. Cette méthodologie est composée de quatre phases pour développer un système multi-agents : analyse préliminaire des besoins, analyse avancée des besoins, conception architecturale et conception détaillée.

- 1) analyse préliminaire des besoins : comprendre le problème en étudiant ses configurations organisationnelles, faire une analyse orientée-but et modéliser les intentions de chaque acteur comme buts. Les acteurs étant des entités sociales dépendantes les unes des autres pour exécuter des tâches, fournir des services et atteindre des objectifs, ces dépendances et relations sont représentées dans un modèle stratégique de dépendance.
- 2) analyse avancée des besoins : décrire le système dans son environnement opérationnel, construire un modèle stratégique rationnel décrivant comment réaliser les buts identifiés dans l'étape précédente, ce modèle est un graphe contenant quatre types de nœuds: objectif, tâche, ressource et objectif affiné, et deux types de liens : décomposition et méthode.
- 3) conception architecturale : définir l'architecture globale du système en termes de sous-systèmes interconnectés via les données, le contrôle et les dépendances,
- 4) conception détaillée : définir le comportement de chaque composant et les communications entre les agents. Tropos adopte des langages de communication existant comme FIPA-ACL et AUML diagrammes.

L'analyse est basée sur le langage de spécification Formal Tropos supportant toutes les notions mentales de Tropos et enrichie par KAOS (van Lamsweerde, 2001).

Limitations:

Tropos manque d'outils permettant le passage de chaque étape à l'étape suivante.

4-3- MaSE

MaSE (Multiagent Systems Engineering) est une approche détaillée d'analyse et de conception des SMA, elle utilise des modèles graphiques dérivés des modèles UML standard pour décrire les types d'agents dans un système et leurs interfaces aux autres agents. MaSE montre les SMA comme une extension du paradigme orienté-objet où les agents sont des objets spécifiés. MaSE est aussi la base de "agentTool", un outil graphique de développement des SMA supportant toutes les étapes de MaSE.

Cette méthodologie est composée de deux phases : une phase d'analyse et une phase de spécification, chaque phase consiste en plusieurs étapes, voir figure3-3.

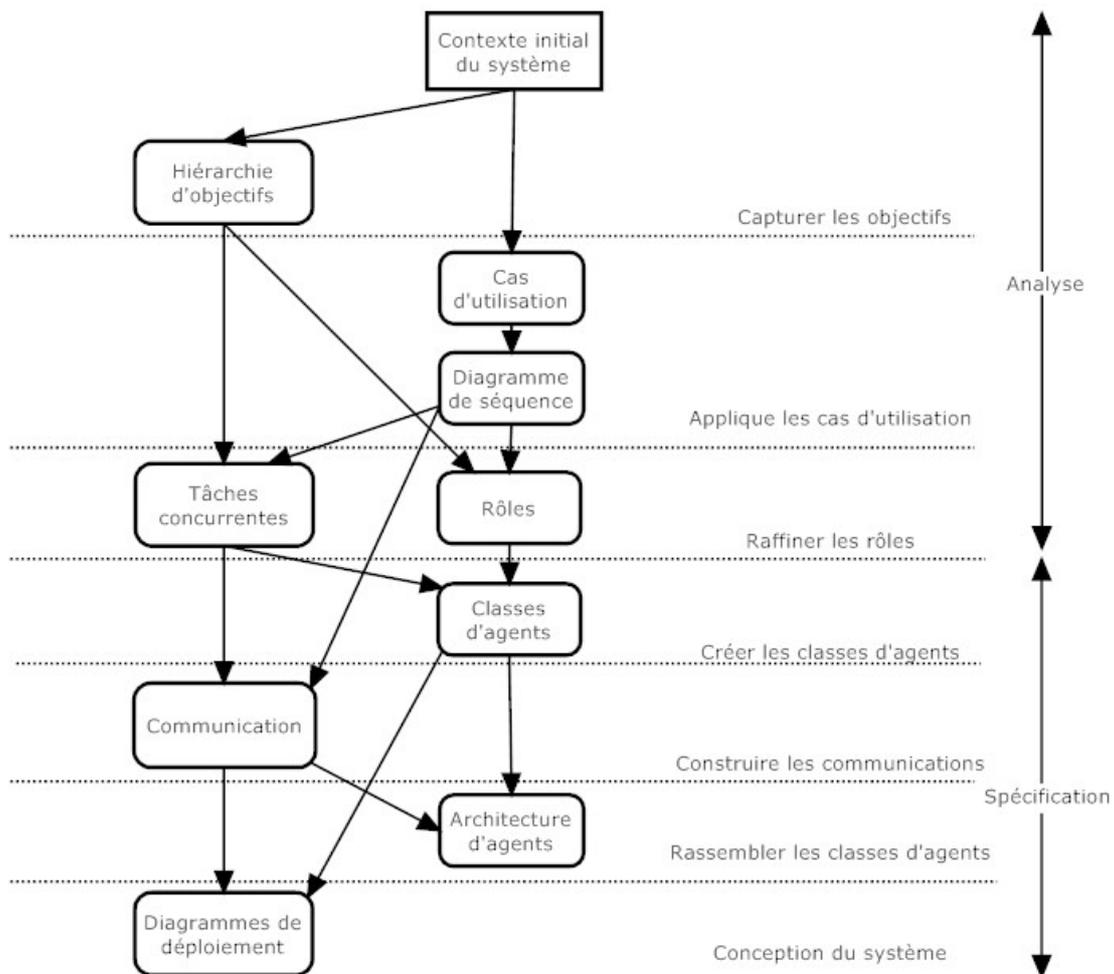


Figure 3-3 : Les phases de MaSE

Des modèles précis sont associés à chaque étape comme le montre le tableau suivant :

Phases	Modèles
1) La phase d'analyse	
- capturer les objectifs	Hiérarchie des objectifs
- appliquer les cas d'utilisation	Cas d'utilisation, diagrammes de séquence
- raffiner les rôles	Tâches concurrentes, modèles de rôle
2) La phase de conception	
- construire les classes d'agent	Diagrammes de classes d'agent
- construire les communications	Diagrammes de communications
- rassembler les classes d'agent	Diagrammes d'architecture d'agents
- conceptualiser le système	Diagrammes de déploiement

Tableau 3-1 : Les modèles associés aux étapes de la méthodologie MaSE.

4-3-1- La phase d'analyse

Cette étape produit un ensemble de tâches et de rôles décrivant comment le système va satisfaire ses objectifs. Les objectifs sont des abstractions des besoins détaillés, ces objectifs sont à réalisés par les rôles. Normalement, un système a un objectif global et un ensemble des sous-objectifs nécessaires pour atteindre l'objectif globale, les objectifs identifient ce que le système essaie d'atteindre (le *Quoi*).

Capter les objectifs :

- L'étape de capture des objectifs commence par l'identification de l'objectif essentiel du système à partir des besoins initiaux. Plusieurs scénarios sont extraits de la spécification initiale, l'objectif de chaque scénario est défini.
- Structurer les objectifs dans un diagramme hiérarchique montrant l'objectif global et les sous-objectifs. Finalement, les sous-objectifs sont décomposés en fonctions décrivant *comment* réaliser l'objectif père.

La décomposition des objectifs n'est pas simplement une décomposition fonctionnelle, les objectifs décrivent le *QUOI* et les fonctions décrivent le *COMMENT*.

On peut trouver quatre types d'objectifs :

- L'objectif *sommaire* qui est dérivé à partir d'un ensemble d'objectifs,
- Des objectifs *partitionnés* où un objectif a des sous-objectifs (comme l'objectif sommaire)

- Des objectifs combinés où plusieurs objectifs similaires sont regroupés dans un seul.
- Des objectifs non-fonctionnels : que sont des objectifs critiques mais ne supportant pas l'objectif global.

La figure 3-4 montre une hiérarchie générale des objectifs :

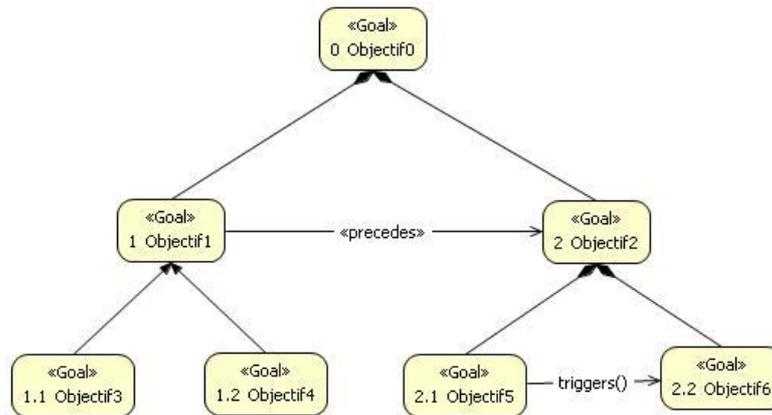


Figure 3-4 : une hiérarchie générale des objectifs dans MaSE

Les relations hiérarchiques entre les objectifs sont des relations AND ou OR, la notation d'agrégation d'UML est utilisée pour présenter le raffinement AND, et la notation de généralisation d'UML est utilisée pour présenter le raffinement OR. Dans la figure précédant, l'objectif *Objectif0* peut être réalisé si les deux objectifs *Objectif1* et *Objectif2* sont réalisés. L'objectif *Objectif1* peut être réalisé si l'objectif *Objectif1* ou l'objectif *Objectif2* est réalisé. Un objectif peut être réalisé avant ou après un autre objectif (*Objectif1* doit être réalisé avant *Objectif2*); la réalisation d'un objectif peut déclencher un autre objectif (*Objectif5* déclenche *Objectif6*). Si un objectif n'a pas un déclencheur, il sera réalisé suite à l'initialisation du système. Un déclencheur est représenté par une flèche entre deux objectifs avec le nom de l'événement déclenchant et l'ensemble des paramètres : *événement* (p_1, \dots, p_n).

Appliquer les cas d'utilisation :

Les cas d'utilisation sont obtenus à partir du cahier de charge pour définir le comportement du système en séquence d'événements, montrant comment le système doit réagir. Ces cas d'utilisations sont ensuite transformés en diagrammes de séquence.

Il existe deux types de cas d'utilisation : positif, décrivant ce que le système doit faire pendant son fonctionnement normal, négatif, décrivant le comportement des cas anormaux.

Raffiner les rôles :

- La création du modèle de rôle : à partir du diagramme hiérarchique des objectifs, les objectifs sont transformés en rôles et tâches associés. Chaque objectif doit être associé à un rôle, et chaque rôle doit être joué par un agent. MaSE ne modélise pas les interactions utilisateur-ordinateur, mais on peut créer un rôle spécifique représentant l'interface utilisateur, les autres ressources comme des fichiers, des bases de données,...etc.

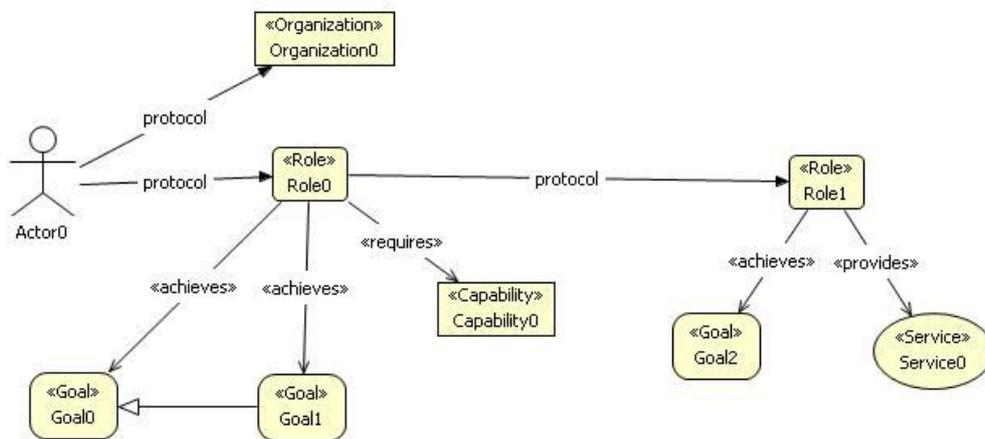


Figure 3-5 : Exemple d'un modèle de rôles dans MaSE

Dans l'exemple de la figure 3-5, Le rôle *Role0* accomplit les objectifs *Goal0* et *Goal1*, et il exige la capacité *Capability0*. Le rôle *Role1* accomplit l'objectif *Goal2* et fournit le service *Service0*. La relation entre *Goal0* et *Goal1* est une relation de précédence (*Goal0* précède *Goal1*).

Des protocoles gèrent les relations entre les rôles et les acteurs.

- Le modèle des tâches concurrentes : définissant le comportement de chaque rôle en identifiant les tâches individuelles, un rôle peut consister en plusieurs tâches. Ce modèle représente ces tâches et leurs communications dans un diagramme d'état fini contenant des états et des transitions.

4-3-2- La phase de conception :

Construire les classes d'agents :

Les diagrammes de classes d'agents sont le premier outil de conception dans MaSE, ils sont créés à partir des rôles définis dans la phase d'analyse. Les rôles sont associés à des types d'agent spécifiques. Les communications entre les classes d'agents sont identifiées dans cette étape, ces communications sont dérivées à partir des communications des rôles associées aux agents (Figure 3-6).

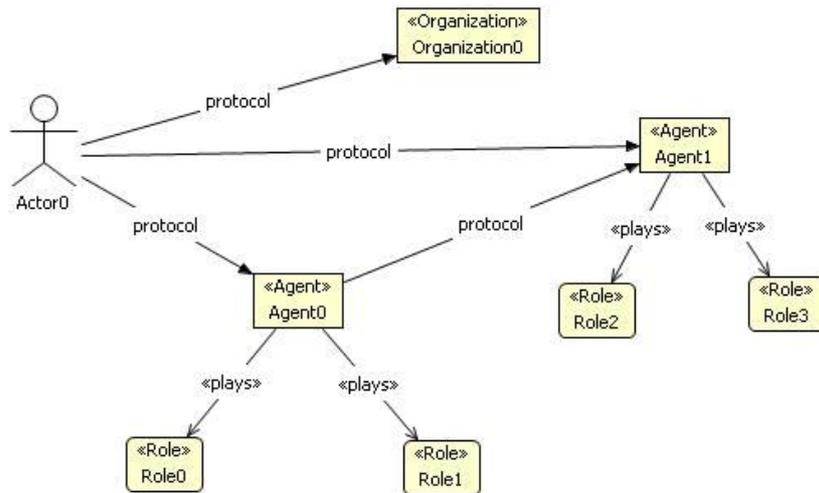


Figure 3-6 : Première possibilité d'un diagramme de classes d'agents

Le diagramme précédent peut être présenté d'une manière plus simple (Figure 3-7) :

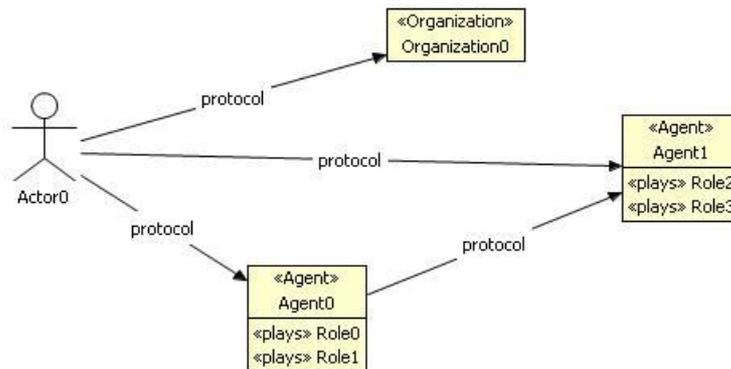


Figure 3-7 : Diagramme de classes d'agents simplifié

Le diagramme de classe d'agent est similaire à un diagramme de classe orienté-objet avec deux différences :

- Les agents sont définis par les rôles qu'ils jouent, au lieu des attributs et des méthodes pour l'objet.
- Les relations entre les classes d'agents représentent les communications.

Construire les communications :

Les communications sont construites à partir des détails internes des tâches concurrentes en utilisant deux diagrammes de communication : un pour l'expéditeur et l'autre pour le récepteur. Ce diagramme est similaire au modèle des tâches concurrentes.

Rassembler les classes d'agent :

Conceptualiser l'architecture interne et le processus de raisonnement de chaque classe d'agents : les développeurs ont le choix entre la conception de leur propre architecture ou l'utilisation des architectures prédéfinies comme l'architecture BDI.

Conceptualiser le système :

Utiliser les diagrammes de déploiement pour définir le nombre, le type et la position des agents dans le système étudié.

4-3-3- AgentTool

AgentTool est un outil graphique de développement des SMA basé sur la méthodologie MaSE. Il possède une interface utilisateur graphique simple, supportant toutes les étapes de MaSE et permettant la vérification des communications et la génération automatique du code dans plusieurs langages de programmation.

4-4- Discussion et comparaisons des méthodologies

Il n'existe pas une façon systématique pour évaluer les méthodologies proposées dans le domaine des SMA, mais en considérant les facettes essentielles d'une méthodologie : les concepts et les propriétés, les notations et les techniques de modélisations, le processus de développement et le pragmatique, on peut constater que les méthodologies présentées précédemment satisfont ces facettes avec un niveau plutôt bon. La méthodologie MaSE donne une description détaillée des activités et du processus de développement, et possède une facilité de passage entre ses différentes étapes, pour ces raisons, MaSE nous semble plus approprié pour l'étude du système de pilotage des systèmes du trafic urbain. Un autre avantage de cette méthodologie est la disponibilité de l'outil agentTool qui supporte complètement toutes les étapes de cette méthodologie.

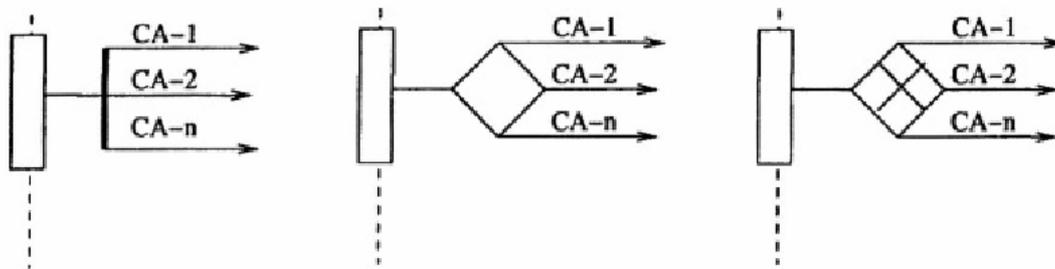
5- Outils et plateformes pour les SMA

Dans cette section nous présentons AUML, langage pour spécifier les composants et le comportement des systèmes étudiés, puis nous présentons deux plateformes de systèmes multi-agents : JADE et Madkit. Nous discutons quelques techniques d'interactions et de coordination et les langages de communication entre les agents.

5-1- L'approche AUML

Il est indispensable d'utiliser des formalismes pour étudier et modéliser des systèmes avec l'approche orientée-agent. Une extension du standard UML appelée AUML (Agent UML) a été proposée dès le début des travaux sur les SMA. L'avantage de cette approche est qu'elle est basée sur UML destinée à la POO et très bien connue dans le domaine de génie logiciel, ce qui permet aux développeurs de passer facilement d'une méthodologie orientée-objet vers l'approche orientée-agent.

AUML a été proposée pour la première fois par (Bauer, 1999), qui a suggéré deux nouvelles spécifications pour pouvoir présenter les interactions entre les agents dans les SMA : le diagramme de séquence avec trois connecteurs AND, OR et XOR, (la figure 3-8 montre ces trois types de connecteurs), et le diagramme de classe d'agent.



a) L'opérateur AND

b) L'opérateur OR

c) L'opérateur XOR

Figure 3-8 : types de connecteurs dans AUML

Nous détaillons ces deux spécifications.

1) Les diagrammes de séquence :

Appelés aussi diagrammes de protocole, leur objectif est de montrer l'échange de messages entre les agents dans le temps en utilisant les protocoles de communication. Ces diagrammes sont composés de deux axes : un vertical représentant le temps, et un

horizontal représentant différents agents ou différents rôles d'agents. La figure 3-9 montre l'échange des messages entre deux rôles:

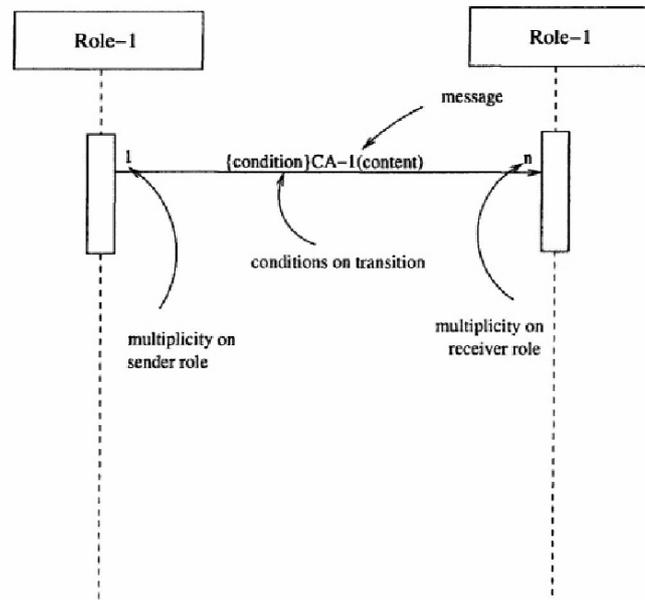


Figure 3-9 : diagramme de séquence et échange de messages dans AAML

Dans les diagrammes de séquences AAML il est possible d'envoyer un message à plusieurs agents ou rôles au même temps.

2) Les diagrammes de classes d'agent :

Les diagrammes de classes UML montrent un ensemble de classes avec leurs attributs, leurs opérations ou les méthodes et les relations entre les différentes classes. Considérant la différence entre un objet et un agent, ces diagrammes doivent être modifiés pour qu'ils correspondent aux caractéristiques des agents. Les diagrammes résultants sont appelés des diagrammes de classes d'agent. Ils contiennent les éléments suivant : le nom de l'agent, une description d'état (vs attributs), les rôles (actions, méthodes, capacités et protocoles), la position organisationnelle de l'agent (sa représentation dans son groupe).

5-2- FIPA et la plateforme JADE

FIPA (Foundation for Intelligent Physical Agents) est une association internationale d'entreprises et d'organisations ayant pour objectif de produire des spécifications pour les technologies génériques des SMA. Les spécifications proposées par FIPA sont essentielles pour tout travail basé sur l'approche agent. Deux rôles principaux ont été suggérés pour une plateforme des SMA (Figure 3-10) :

- Un système de gestion des agents, Le AMS « Agent Management System » qui surveille les agents, leur authentification, leur accès et l'utilisation du système.
- Le facilitateur de répertoire DF « Directory Facilitator » fourni un service de *pages jaunes* à la plateforme.

FIPA a aussi proposé des spécifications pour un langage de communication des agents : ACC (Agent Communication Chanel) basé sur l'envoi de messages.

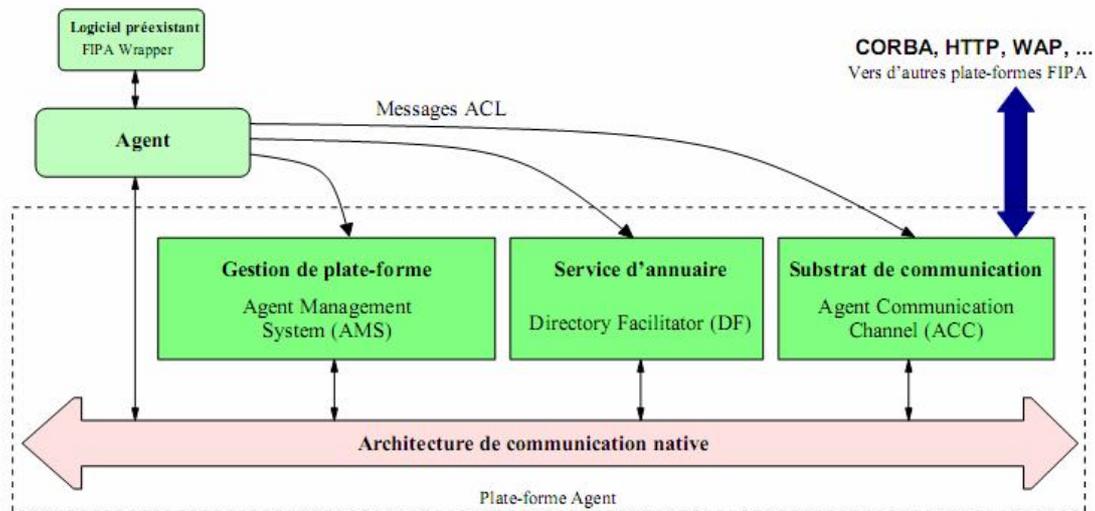


Figure 3-10 : Un modèle de référence des plateformes SMA dans FIPA

JADE (Java Agent DEvelopment framework) est un environnement logiciel pour développer des applications SMA en conformité avec les spécifications FIPA. Malheureusement, il manque une méthodologie de développement. Jade fourni des classes qui implémentent «JESS» pour la définition du comportement des agents. L'outil possède les trois modules proposés par les spécifications FIPA. Les agents communiquent grâce au langage FIPA ACL. Un éditeur est disponible pour l'enregistrement et la gestion des agents. Aucune autre interface n'est disponible pour le développement ou l'implémentation. À cause de cette faiblesse, l'implémentation demande beaucoup d'efforts. Elle nécessite une bonne connaissance des classes et des différents services offerts.

5-3- La plateforme MadKit

MadKit est une plateforme Multi-agent en Java établie sur un modèle d'organisation. Il est développé par Olivier Gutknecht et Jacques Ferber au LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier).

MadKit est la plupart du temps un moteur d'exécution de SMA, en utilisant un micro-noyau d'agents. Le modèle d'organisation fondamental est appelé Aalaadin.

En plus de ce modèle d'organisation, Madkit est basé sur trois principes :

- Une architecture à micro noyau
- Une agentification systématique des services
- Un découplage fonctionnel entre noyau, agents et application d'accueil

Madkit est un ensemble de packages Java qui implémentent le noyau agent, diverses bibliothèques de base de messages, d'agents et de sondes. La figure 3-11 montre la structure générale de Madkit (Gutknecht *et al*, 1999), (Gutknecht, 2001), (Ferber, 1999, 2004) :

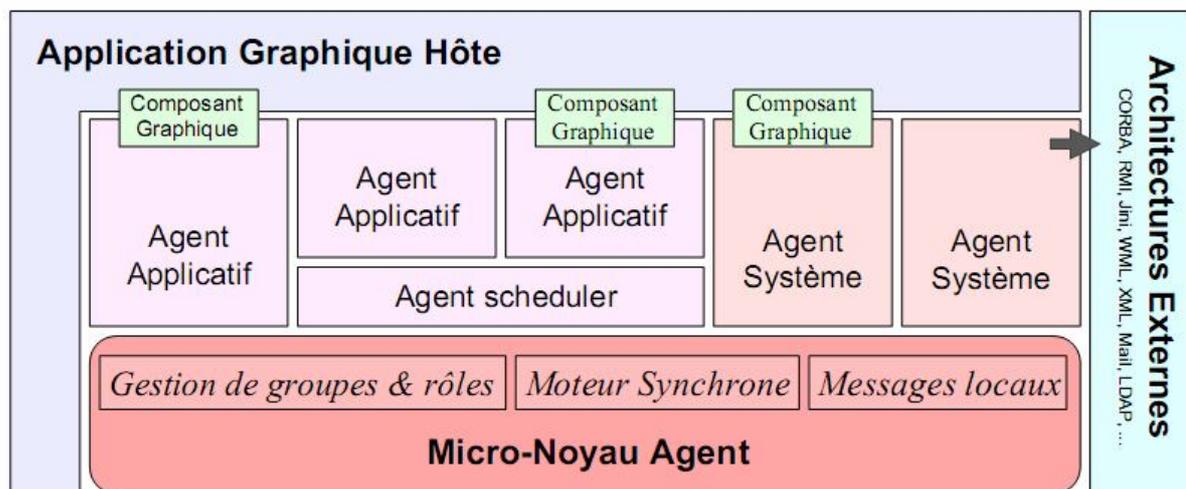


Figure 3-11 : Structure générale de Madkit

Le micro-noyau est un environnement d'exécution d'agents de taille réduite qui prend en charge les fonctions suivantes :

- Gestion de groupes et rôles locaux
- Gestion de cycle de vie des agents
- Passages des messages locaux
- Observation et exécution

La classe de base d'un agent Madkit "*AbstractAgent*" définit les fonctionnalités suivantes

- Cycle de vie : l'agent dispose de quatre états (création, activation, exécution et destruction) et d'un dispositif d'activation d'autres agents.

- Communication : elle est implémentée sous forme de passage de messages asynchrones soit d'un agent à un autre identifié par ses *AgentAdress* ou par son groupe et rôle, soit sous la forme d'une diffusion à tous les teneurs d'un rôle dans un groupe donné.
- Organisation : chaque agent dispose de primitives lui permettant d'observer son organisation locale et d'y agir.
- Outils : chaque agent peut manipuler une interface graphique qui lui est associée et les flots d'entrée/sortie ... etc.
- Messages : Les messages sont hérités à partir d'une classe de base "*Message*" qui définit la notion d'émetteur et de destinataire. Une bibliothèque de messages est fournie et permet l'envoi de chaînes, d'objets sérialisés, de documents XML, ou de messages conformés aux spécifications KQML et FIPA-ACL (voir figure 3-12)

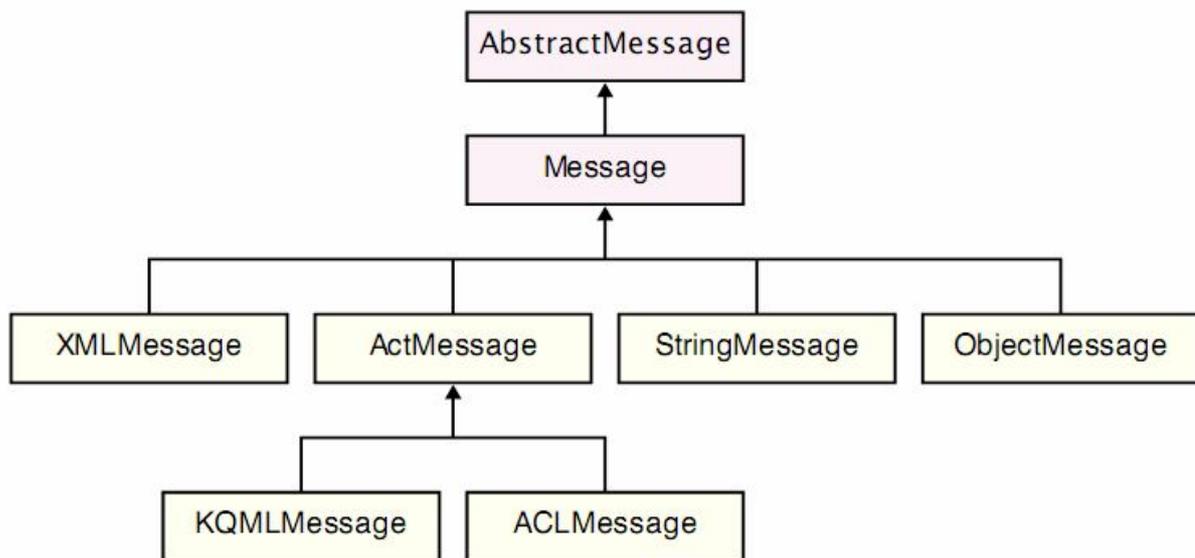


Figure 3-12 : Hiérarchie des messages standards

Aucune méthode spécifique d'analyse n'est associée à MadKit. Cette étape devrait inclure une analyse fonctionnelle, une analyse de dépendance, une découverte de contexte de groupe et un choix des mécanismes de coordination. N'importe quelle méthode d'analyse du domaine peut être employée.

L'étape de conception dans MadKit inclut la définition du modèle d'organisation (groupes, rôles), du modèle d'interaction (protocoles, messages), et d'autres entités spécifiques (tâches, buts, etc..). Le modèle d'Aalaadin donne des indications pour réaliser la conception, mais aucun outil logiciel n'est fourni. En dépit de l'orientation organisationnelle du modèle d'Aalaadin, il est en fait applicable à une large gamme de

conception de SMA. L'accomplissement de cette étape est facilité en utilisant les notions de groupes et de rôles. En plus, le groupe et les définitions de rôles peuvent être réutilisés. La figure 3-13 montre un modèle Aalaadin (AGR) étendu (Michel, 2004), (Mansour, 2007) :

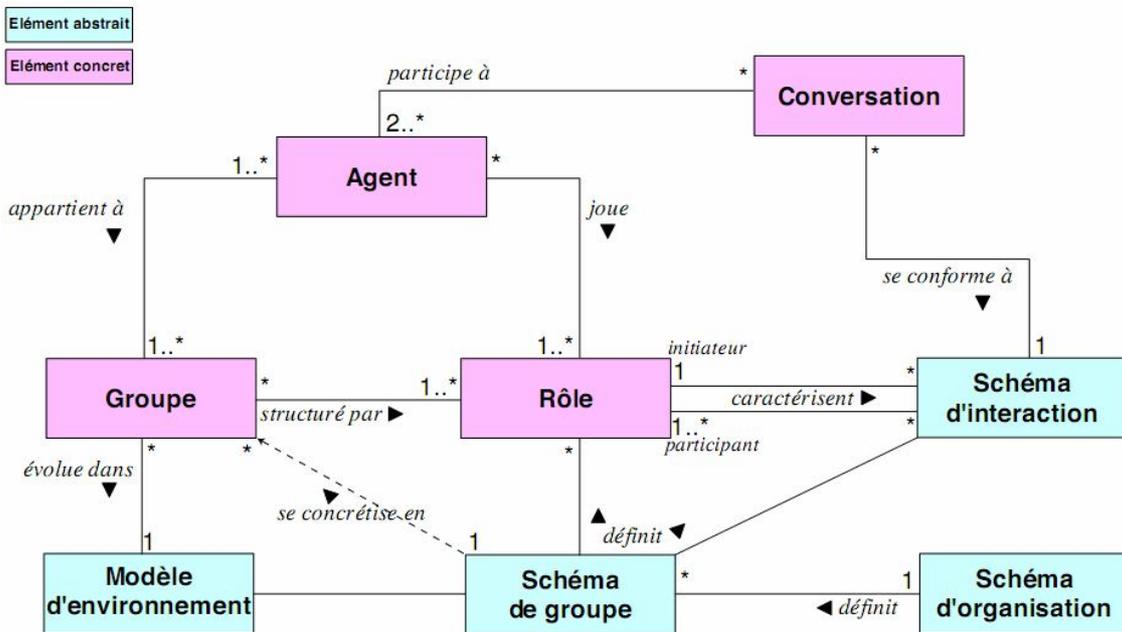


Figure 3-13 : Modèle AGR étendu

Nous définissons succinctement les trois notions : agent, groupe et rôle :

Agent : Le modèle AGR n'impose pas d'architecture interne de l'agent. *Un agent* est seulement indiqué comme entité active de communication qui joue « *des rôles* » dans « *des groupes* ». Le concepteur peut donc adopter la définition qui lui convient.

Groupe : les *Groupes* sont définis en tant qu'ensembles atomiques d'agrégation d'agents. Chaque agent fait partie d'un ou plusieurs groupes. Sous sa forme plus fondamentale, le groupe est seulement une manière d'étiqueter un ensemble d'agents. Les groupes ont les caractéristiques suivantes :

- Un agent peut être un membre *de n* groupes au même instant.
- Les groupes peuvent se recouvrir.
- Un groupe peut être créé par n'importe quel agent, et un agent peut demander son admission dans n'importe quel groupe.
- Avec les agents additionnels de la plate-forme, un groupe peut être local ou distribué sur plusieurs machines.

Rôle : Le rôle est une représentation abstraite d'une fonction, d'un service de l'agent. Chaque agent peut assurer des rôles multiples, et chaque rôle d'un agent est local à un groupe.

L'étape de développement inclut le choix du modèle d'agent, son exécution, et l'exécution des stratégies de protocole d'interaction. Aucun modèle d'agent n'est fourni, il doit être mis en œuvre avec le langage Java. Puisque aucun modèle d'agent n'est préconisé, n'importe quel type de modèle peut être employé.

Le déploiement des agents de MadKit est réalisé dans la G-boîte, où des agents peuvent être créés, modifiés et détruits, avec une interface graphique. Plusieurs G-boîtes peuvent être reliées pour réaliser la distribution à travers un réseau. Des agents sont empaquetés dans des archives de type Jar, et peuvent être mélangés avec d'autres agents, permettant un niveau élevé de réutilisation d'agent.

Pour clarifier les relations entre noyau, agent, et les applications hôte, la figure 3-14 montre la séquence des échanges entre ces composants pendant l'amorçage de Madkit (Gutknecht, 2001) :

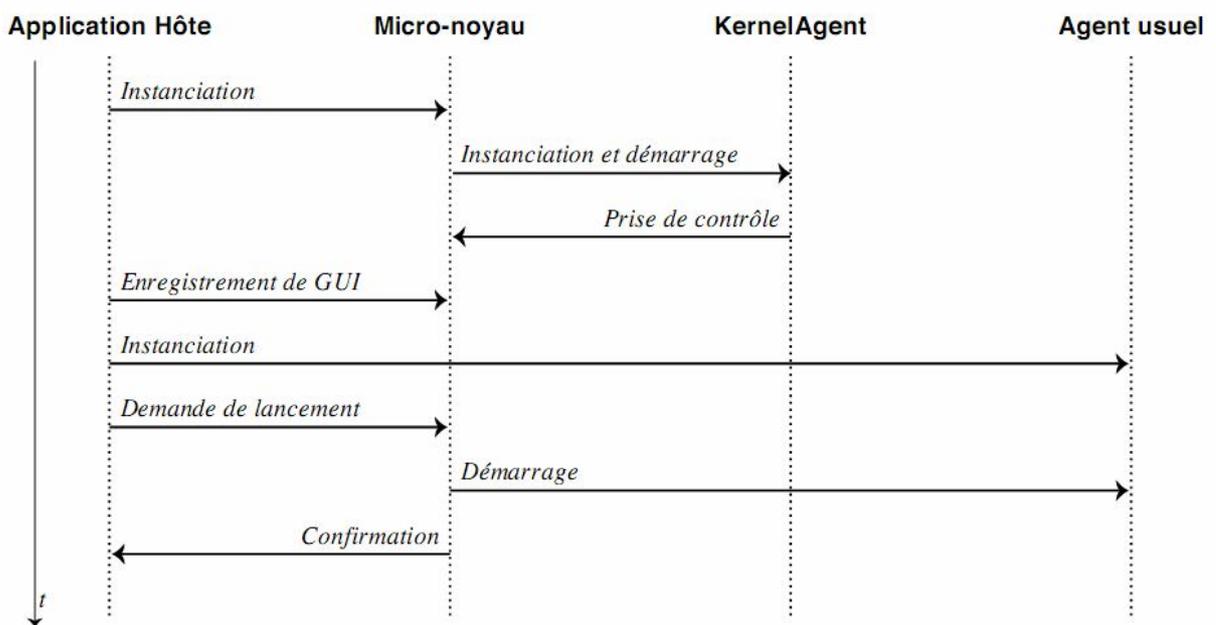


Figure 3-14 : Amorçage de madkit

MadKit est également une plateforme distribuée qui tient compte du développement des applications réparties efficaces. Pour les programmeurs, les composants distribués de base tels que le « sockets » et « ports », sont totalement

transparents. Une application développée en utilisant une méthode multi-agents peut fonctionner d'une manière distribuée sans changer une ligne de code.

MadKit est libre pour l'usage éducatif. MadKit est toujours en évolution, afin d'ajouter de nouveaux dispositifs.

La caractéristique principale de MadKit est que c'est la plupart du temps un moteur d'exécution Multi-agents. Le développement et le déploiement sont simples, puisque la plateforme se concentre sur l'infrastructure des agents. MadKit manque d'une méthodologie indépendamment du modèle d'Aalaadin, qui est plus un modèle d'organisation descriptif qu'une méthode de conception. Le défaut principal de MadKit, est que la construction des agents complexes exige beaucoup d'écriture de code, puisqu'il n'y a aucun modèle préétabli d'agent. Cependant n'importe quel programmeur peut commencer à écrire ses propres agents, même sans une bonne connaissance des SMA, ce qui rend MadKit très intéressant dans un contexte éducatif.

6- Interactions, communications et coordination

Infrastructures et techniques de coordination :

La coordination est un élément critique dans les SMA. Nous pouvons définir la coordination comme une gestion des dépendances des agents dans un SMA, dont l'objectif est d'intégrer plusieurs activités exécutées par plusieurs agents pour réaliser un objectif en commun ou un fonctionnement global.

Il est nécessaire de différencier plusieurs notions : coordination, coopération et co-construction :

- Coordination : une activité collaborative pour gérer les interactions et les dépendances entre les agents. Les agents partagent leur environnement ou une partie de cet environnement, mais ils se concentrent sur leurs objectifs.
- Coopération : les différents agents ont un objectif en commun et ils réagissent et ils peuvent influencer les uns les autres pour réaliser cet objectif.
- Co-construction : des interactions qui exigent un changement organisationnel de l'agent et de ses interactions envers les objets partagés et l'environnement (identifier les objectifs et les tâches sociales).

La compréhension de ces notions est nécessaire pour pouvoir réaliser les différents types de communication entre les agents dans leur environnement.

7- Conclusion

Nous avons présenté l'approche multi-agents comme une approche récente et utile pour étudier les systèmes complexes ou distribués. Connaître toutes les méthodologies et les outils existants basés sur cette approche exige un très grand effort, nous nous sommes concentrés sur les trois méthodologies les plus connues et les plus génériques : Gaia, Tropos et MaSE. Dans le but d'étudier et de conceptualiser le sous-système décisionnel des STU, nous avons choisi, en justifiant ce choix, MaSE pour la conception de ce sous-système. Cette méthodologie est simple et détaillée et donne la possibilité de passer d'une étape à une autre efficacement. L'existence d'un outil de conception agentTool basé sur cette méthodologie est un autre avantage.

Nous avons présenté les spécifications FIPA, et les éléments clés pour les plateformes des SMA. Deux plateformes : Jade et Madkit qui respectent ces spécifications ont été présentées.

Madkit est une plateforme d'implémentation basée sur le paradigme AGR (Agent, Group, Rôle), l'adoption de ce paradigme qui est similaire à l'organisation des STU facilite la conception des entités décisionnelles.

Dans le chapitre suivant nous présentons l'utilisation de l'approche multi-agents et de la méthode MaSE pour la conception de la partie décisionnelle des STU dans le cadre de la méthodologie ASCI-mi STU. Nous présenterons l'implémentation de notre système multi-agents sous la plateforme Madkit.

Chapitre 4

- Proposition d'une approche agent pour modéliser le Sous-Système Décisionnel des STU

Sommaire

1- INTRODUCTION.....	81
2- ASCI-MI UNE MÉTHODOLOGIE MULTIPLE ET INCRÉMENTIELLE POUR LES STU	81
3- L'APPROCHE MULTI-AGENTS DANS LE CONTEXTE DU SOUS-SYSTÈME DÉCISIONNEL SSD	92
4- MASE POUR LA CONCEPTION DU SSD	93
<i>4-1- Analyse.....</i>	<i>94</i>
<i>4-2- Conception</i>	<i>102</i>
5- CONCLUSION	107

1- Introduction

Ce chapitre est dédié à l'étude du sous-système décisionnel (SSD) des STU et de ses entités de régulation, la conception de ces entités revient à la conception du processus décisionnel. Pour cet objectif nous utilisons MaSE qui est basée sur les systèmes multi-agents. Le point de départ est les travaux réalisés par (Sarramia, 2002). La démarche de MaSE sur le pilotage des STU sera détaillée dans le cadre de la méthodologie ASCImi-STU proposée par (Sarramia, 2002) afin d'identifier les objectifs et les rôles des entités décisionnelles et de construire le modèle de classes d'agents. L'intérêt de ASCImi-STU est d'avoir une démarche méthodologique qui permet de modéliser l'ensemble du domaine et de relier la partie décisionnelle avec les autres parties (les parties physique et logique).

Nous présentons en premier temps la méthodologie ASCImi-STU, ensuite nous détaillons les phases d'analyse et de conception du SSD en utilisant MaSE.

2- ASCI-mi une méthodologie multiple et incrémentielle pour les STU

Considérant les complexités structurelle et fonctionnelle des systèmes du trafic urbain et les interactions fortes entre ses sous parties, la méthodologie ASCI (Analyse, Spécification, Conception et Implémentation) a été adoptée. Cette méthodologie a été proposée par (Gourgand, 1991), elle a été utilisée dans différents domaines comme les systèmes hospitaliers (Combes, 1994), (Fenies, 2006); les systèmes industriels (Goujon, 1998), (Lacomme, 1998), et (Tchernev, 1997); et également les systèmes du trafic urbain (Sarramia, 2002).

ASCImi est une méthodologie de conception et d'implémentation d'environnements de modélisation et de simulation d'une classe de systèmes comme la classe des STU. Une classe de systèmes regroupe des systèmes ayant des caractéristiques techniques et fonctionnelles semblables. **ASCI**mi doit permettre la construction de modèles pour tout système d'une même classe.

ASCImi préconise la construction d'un modèle générique et unique de connaissance donnant une vue complète du domaine. Le modèle générique de connaissance est ensuite traduit sous forme de composants logiciels.

La modélisation du domaine fournit deux types de composants : les composants de modélisation et les composants logiciels. La modélisation d'un système appartenant au domaine est basée sur le processus itératif de modélisation suivant :

- L'élaboration d'un modèle de connaissance d'un système à partir du modèle générique de connaissance du domaine.
- L'obtention d'un modèle d'action à partir du modèle de connaissance et de la bibliothèque de composants logiciels.
- L'exploitation du modèle d'action pour évaluer les performances du système.
- L'interprétation des résultats, correspondant à la déduction des actions à effectuer sur le système.

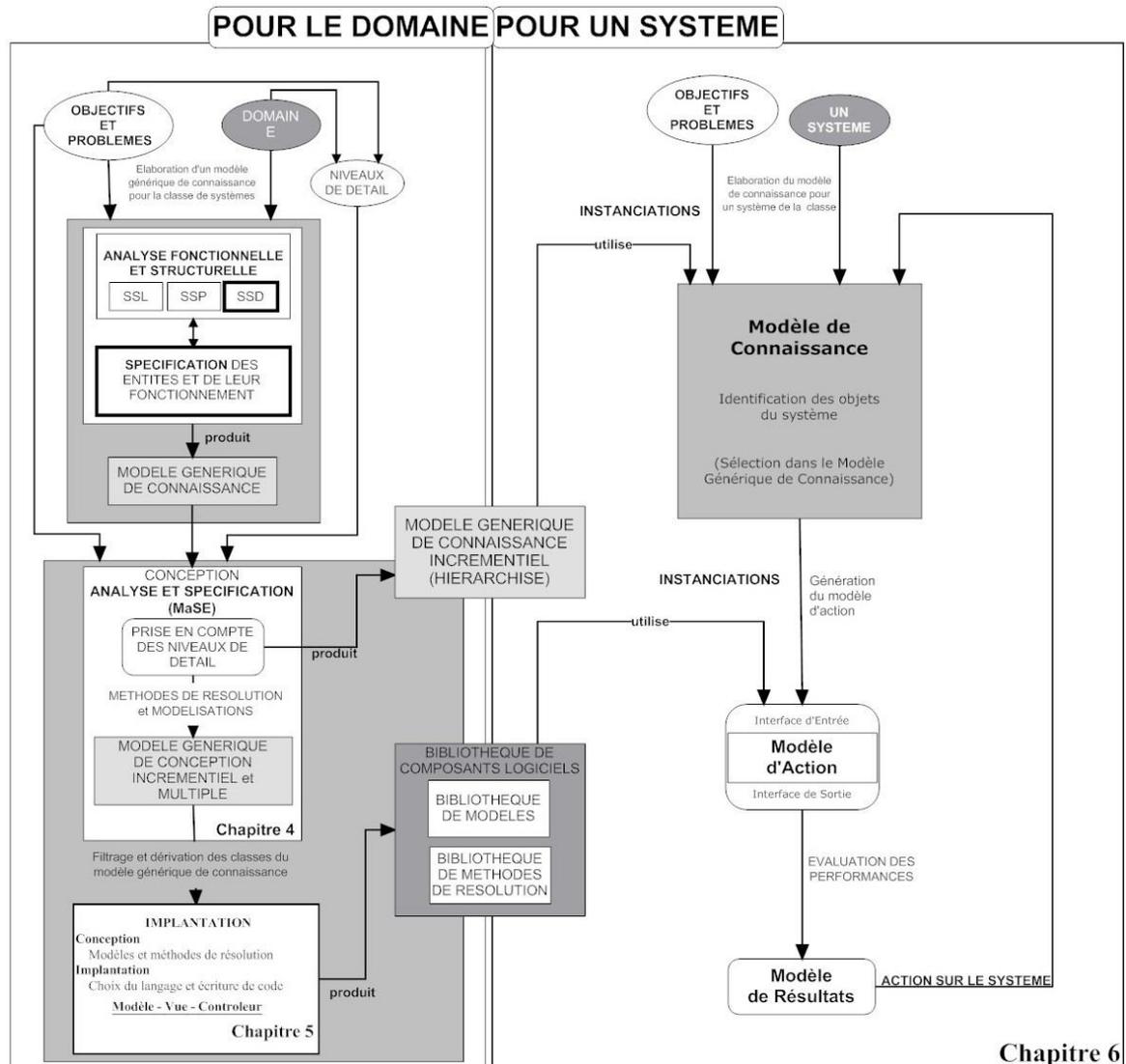


Figure 4-1 : La méthodologie de modélisation ASCI-mi

A partir de la figure 4-1 illustrant ce processus, nous pouvons distinguer deux parties : la partie illustrée à gauche est relative au domaine étudié, et la partie droite correspond au système étudié (existant ou à concevoir).

L'objectif de la première partie est de capitaliser la connaissance acquise en termes de modèle de connaissance et de composants par l'étude successive de systèmes appartenant au domaine. Ce modèle de connaissance et les composants seront utilisés pour instancier dans la deuxième partie, le modèle d'action du système étudié.

Les deux parties sont mises en œuvre au cours de l'étude d'un système et sont composées de quatre phases :

Analyse

La méthodologie ASCI permet d'analyser et de comprendre le fonctionnement du domaine en fonction de l'objectif à atteindre. La phase d'analyse est basée sur une décomposition du système étudié en trois sous-systèmes disjoints mais communicants afin de faciliter la construction du modèle de connaissance :

- Le **sous-système logique** noté **SSL** est constitué des entités que le système doit traiter.
- Le **sous-système physique** noté **SSP** définit l'ensemble des moyens physiques et l'infrastructure.
- Le **sous-système décisionnel** noté **SSD** contient les règles de gestion et de fonctionnement du système.

Le SSD est le sous-système le plus complexe. Il agit à la fois sur le sous-système physique et le sous-système logique et il assure les activités suivantes :

- Réception des informations des autres sous-systèmes (par exemple par le biais de capteurs).
- Action sur le SSL : i.e. mise en œuvre de règles de choix d'allocation des ressources.
- Action sur le SSP : i.e. mise en place de règles de fonctionnement de l'attribution des ressources.

La figure 4-2 montre un exemple de communications entre les trois sous-systèmes tiré des flux urbains :

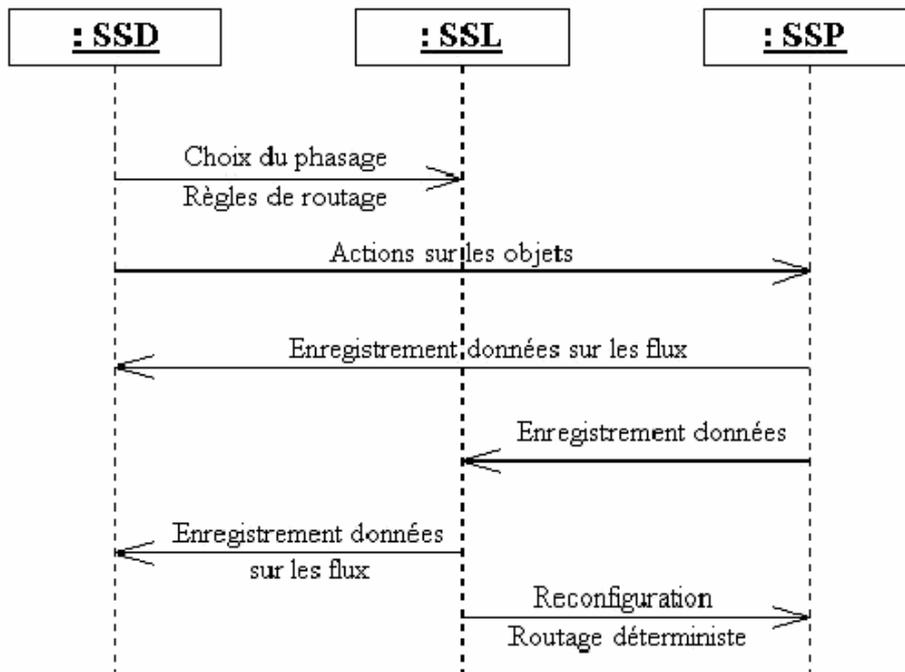


Figure 4-2 : Diagramme de séquence présentant les communications entre les trois sous-systèmes

Un modèle générique de chaque sous-système est réalisé et leurs communications sont décrites, chaque sous-système est représenté en utilisant les diagrammes de classe UML. Ces diagrammes capturent les aspects statiques du domaine.

Spécification

En raison de la nature dynamique et de la complexité des STU, il est nécessaire d'utiliser des diagrammes supplémentaires pour présenter l'aspect dynamique. Il est possible d'utiliser les *cas d'utilisation* et les *diagrammes de séquence* pour décrire le processus et la hiérarchie de prise de décision. Les *diagrammes de séquence* permettent également d'exprimer l'échange de données entre les trois sous-systèmes.

Les étapes d'**Analyse** et de **Spécification** permettent de construire le modèle générique de connaissance, elles sont dépendantes du domaine, mais indépendantes de l'environnement d'implémentation.

Le modèle générique de connaissance obtenu par les étapes d'**analyse** et de **spécification** mentionnées doit être traduit sous la forme de composants logiciels durant les étapes de **Conception** et **Implémentation**.

Conception

Cette phase concerne la partie décisionnelle des STU, elle est découpée en deux étapes: Analyse et Spécification. Nous proposons d'utiliser MaSE présentée dans le troisième chapitre pour cette phase.

Dans l'étape d'analyse, les entités du SSD seront identifiées y compris les entités actives.

Dans l'étape de spécification, l'approche multi-agents exige l'utilisation de diagramme de classe d'agent au lieu du diagramme de classe (utilisé dans l'approche orientée objet).

Implémentation

Le modèle d'implémentation correspond au choix du langage de programmation et au codage effectif des composants logiciels. Les outils et la plateforme utilisés pour implémenter notre système multi-agents seront choisis dans cette étape, ainsi que le modèle d'agent qui est utilisé comme base pour nos agents sera déterminé.

Notions sur la modélisation multiple et incrémentielle :

Une idée de ASCImi-STU est de mettre en place une architecture logicielle générique et ouverte dans laquelle peut être implémenté n'importe quel modèle. La modélisation **multiple** permet d'obtenir différentes modélisations pour la même entité. Ces modèles sont utilisés soit indépendamment les uns des autres soit en collaboration. En plus de l'aspect multi-modèle, la simulation a été conçue pour prendre en compte les niveaux de détail macroscopique et microscopique ainsi que les passages de l'un à l'autre.

La modélisation **incrémentielle** permet la construction des modèles étape par étape. Les notions de modélisation multiple et incrémentielle sont prises en compte au niveau de la conception des composants logiciels.

Cette méthodologie est un cadre global et générique permettant d'étudier un système et de traiter ses problèmes en partant de l'analyse jusqu'à l'interprétation des résultats.

L'environnement logiciel de modélisation et de simulation construit autour de la méthodologie ASCImi-STU comprend les couches données par la figure 4-3 :

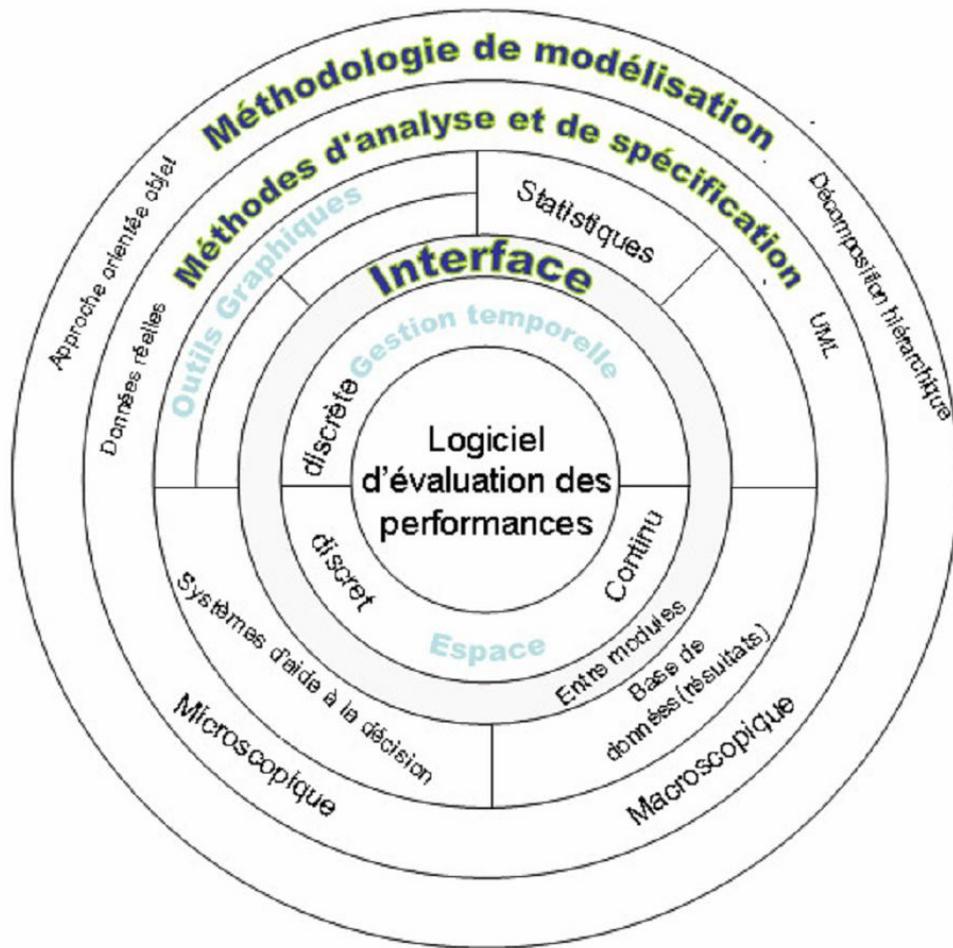


Figure 4-3 : L'environnement ASCImi-STU

Les phases d'analyse et de spécification de la méthodologie ASCImi appliquée aux STU (Sarramia, 2002) ont mené au modèle générique de connaissance du domaine. La décomposition en trois sous-systèmes est faite en premier lieu. Le modèle de chaque sous-système et le modèle de communication entre les trois sous-systèmes sont construits. Nous présentons ces trois systèmes :

Le sous-système logique (SSL) :

Le sous-système logique présenté par la figure 4-4 est composé des entités de flux qui expriment la nature et la composition des flux que les systèmes étudiés doivent traiter. Une entité de flux peut être un flux de véhicules, un flux d'informations ou un flux de décisions. Un trajet est composé d'une origine, une destination, et de points intermédiaires. Un flux de véhicules est effectué à un trajet. Les plans de feux sont exprimés en utilisant un diagramme de feux, un diagramme de

feux est associé à chaque carrefour à feux, il contient l'ensemble des plans de feux pour les feux de l'intersection considérée.

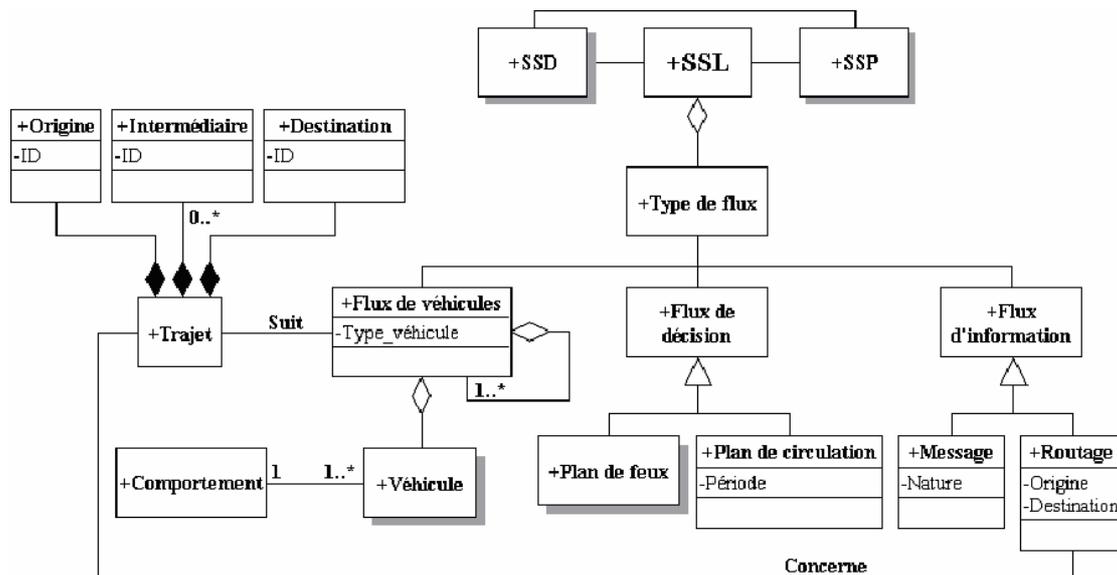


Figure 4-4 : Diagramme de classe du SSL

Le sous-système physique (SSP) :

Le sous-système physique concerne la topologie des STU, cette topologie est exprimée par le diagramme de classe (figure 4-5). Les routes sont connectées entre elles par des intersections, plusieurs types d'intersections sont possibles (rond-point, carrefour à feux, ...). L'ensemble des routes et d'intersections sont composées de voies, de chaussées et de jonctions.

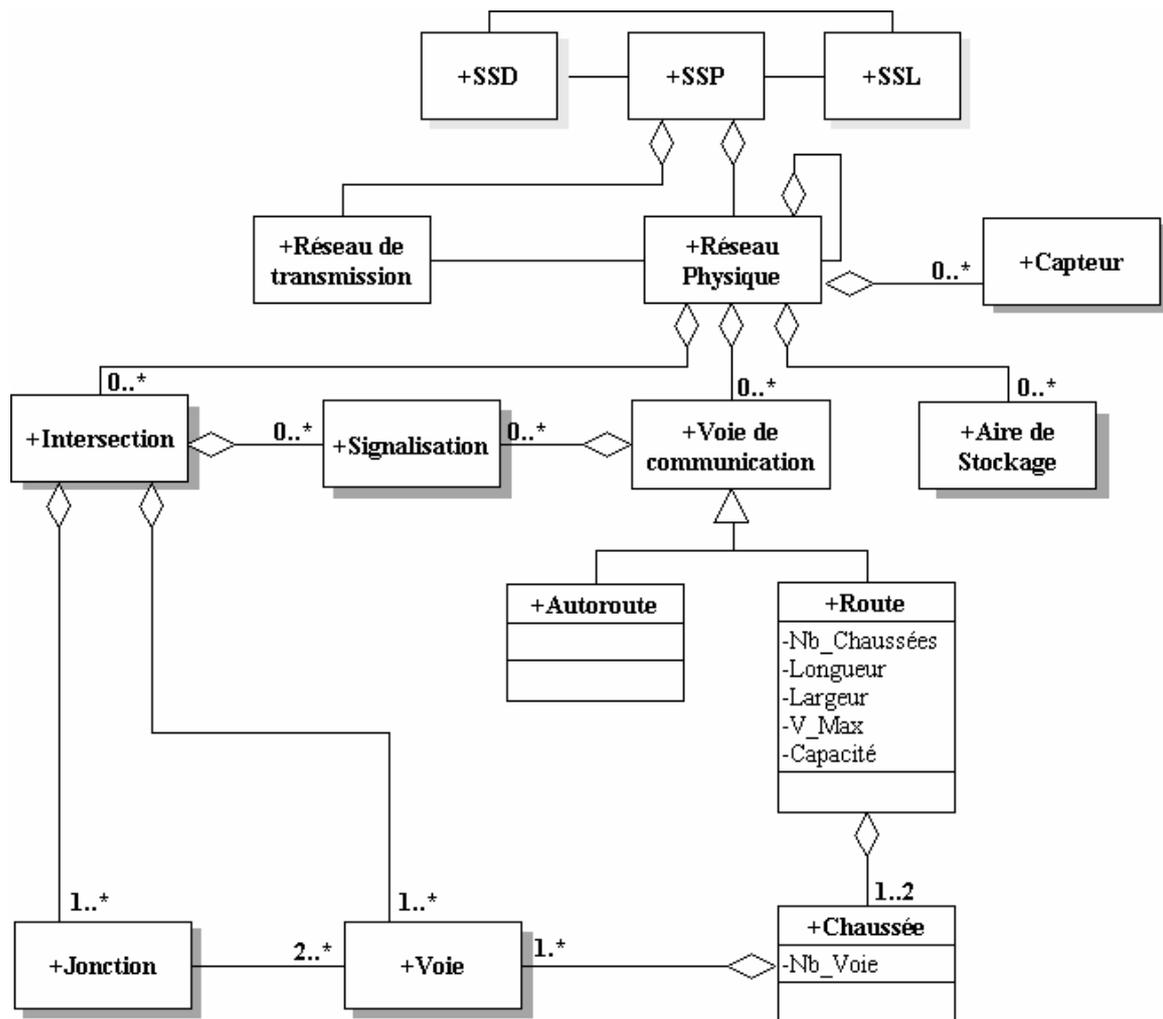


Figure 4-5 : Diagramme de classe du SSP

Le sous-système décisionnel (SSD) :

Les travaux présentés dans (Sarramia, 2002) ont permis d'obtenir le modèle générique de connaissance des STU et de modéliser les deux parties physique et logique des STU, i.e. l'infrastructure et la charge, et d'identifier les entités décisionnelles passives et actives sans en préciser leurs fonctionnements détaillés et leurs communications. Ces entités figurent dans le diagramme de classe 4-6 :

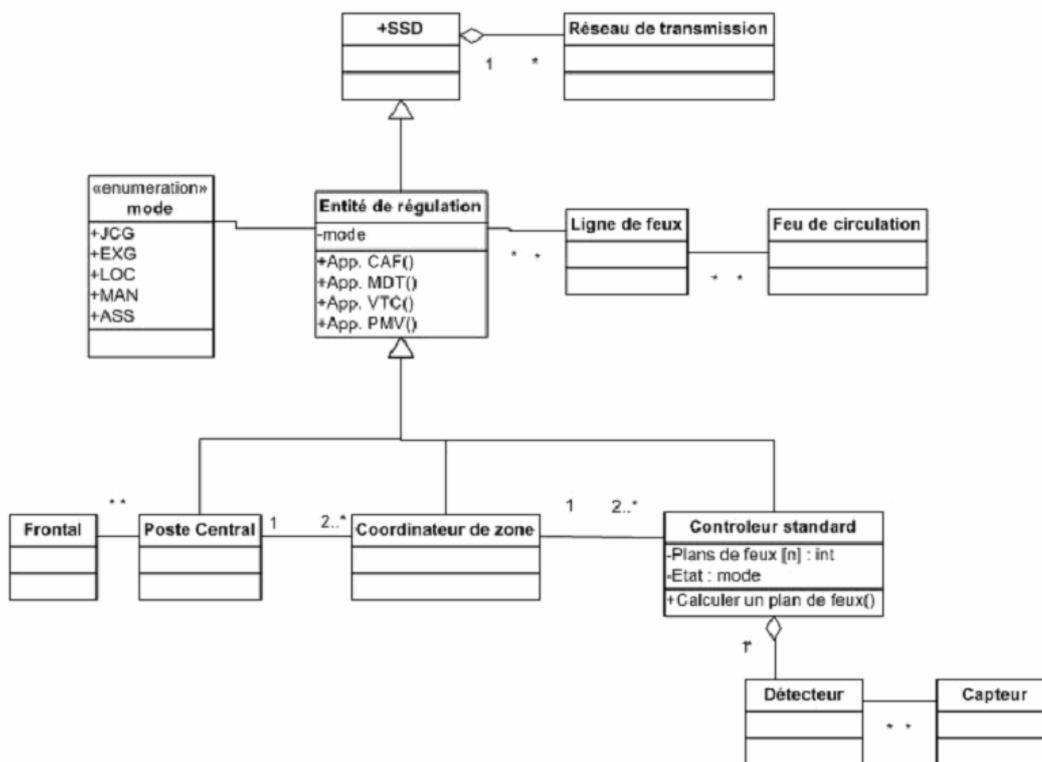


Figure 4-6 : diagramme de classe du SSD.

Nous nous concentrons sur la conception et l'implémentation du sous-système décisionnel des STU pour arriver à un modèle représentant ce sous-système et pour la mise en œuvre d'un outil d'aide à la décision afin de mieux exploiter le modèle de simulation obtenue avec ASCI-mi STU (SSL, SSP) et de mieux interpréter ses résultats.

Considérant les caractéristiques des entités actives du sous-système décisionnel et la complexité des interactions fortes entre elles et avec le décideur, nous adoptons l'approche multi-agents pour construire un modèle du sous-système décisionnel en utilisant :

- MaSE pour les étapes d'analyse et de spécification de la phase de conception.
- Madkit pour la phase d'implémentation.

Nous présentons le sous-système décisionnel des systèmes du trafic urbain, et nous justifions la nécessité d'utiliser l'approche multi-agents pour modéliser cette partie des STU (l'ensemble des informations et la prise de décision sont distribués géographiquement).

L'approche multi-agents semble appropriée pour étudier les systèmes complexes et les systèmes distribués tel que le pilotage du STU. Considérant sa complexité et la structure hiérarchique et décentralisée de son pilotage, l'adoption de cette approche permet de simplifier cette complexité et de mieux comprendre le fonctionnement et le comportement de différentes entités notamment les entités de régulation.

L'approche multi-agents peut être vue comme un couplage entre le paradigme orienté objet et l'intelligence artificielle où les agents sont des objets spéciaux ayant des méthodes qui peuvent être appelées par d'autres agents, et les agents peuvent se coordonner les uns avec les autres en utilisant des moyens de communications, ils peuvent réagir proactivement pour réaliser des buts individuels ou en commun.

Puisque cette étude est dédiée au SSD nous retenons que les entités qui nous intéressent du point de vue décisionnel. Nous ajoutons d'autres entités permettant d'exprimer l'essentiel de la partie décisionnelle.

Le STU est un réseau de routes et d'intersections (carrefours), une route est un ensemble de voies ouvertes à la circulation publique. Une route est composée des chaussées qui peuvent être divisées en plusieurs voies, les voies n'ont pas forcément le même sens de parcours, elles peuvent être matérialisées par des lignes continues ou discontinues. Parfois, les voies sont réservées à certaines catégories de véhicules (autobus, taxi,...).

Une intersection peut être considérée comme une ressource commune entre plusieurs axes, donc la hiérarchie de ces entités est différente de celle présentée par le diagramme de classe du SSP (figure 4-5). Les voies seront regroupées selon leur point d'entrée à une intersection.

La nouvelle hiérarchie et les communications entre les trois sous-systèmes sont représentés sur la figure 4-7 :

3- L'approche multi-agents dans le contexte du sous-système décisionnel SSD

En utilisant la méthodologie ASCI-mi, (Sarramia, 2002) a développé une librairie de composants logiciels. Celle-ci intègre complètement le SSP et le SSL des STU selon une approche orientée objet. Ses travaux ont permis d'identifier les entités actives et passives du SSD, sans pour autant en préciser leur fonctionnement et leur communication. L'étude de ces entités actives revient à spécifier le processus décisionnel.

L'approche multi-agents nous permet de préciser le fonctionnement et la communication des entités de régulation du sous-système décisionnel.

Le STU est un système dynamique, composé de zones géographiquement distribuées. Chaque zone est dirigée par un coordinateur de zone, ce dernier gérant au moins deux contrôleurs de carrefour à feux. De fait, le contrôleur dispose d'un environnement composé, entre autres : de chaussées, de voies, de capteurs, de feux de signalisation et de véhicules. Le tout est lié de façon hiérarchique au poste centrale.

Le diagramme de classe de la figure 4-7 montre les différentes entités du sous-système décisionnel. On y trouve :

Des entités passives : voie, courant, ligne de feux, carrefour à feux, les différentes types de signalisations, et le détecteur.

Des entités actives : Ce sont les entités de régulation : le poste central, les coordinateurs de zone et les contrôleurs standards.

Nous détaillons par la suite les trois entités de régulation qui sont les contrôleurs standards, les coordinateurs de zones et le poste central.

Le Poste Central (PC) a pour rôle de coordonner l'ensemble des autres entités de régulation pour que l'état du système reste stable. Grâce aux informations fournies par les contrôleurs, le PC connaît en permanence l'état actuel du système. Il possède la connaissance (données agrégées ou non selon la granularité utilisée) sur le système complet et il choisit de modifier tel ou tel plan de feux pour améliorer le fonctionnement du système. Il est seul à avoir une vision globale des plans de feux appliqués dans le système. Il assure un contrôle global sur le système.

Le Coordinateur De Zone (CDZ) coordonne un ensemble de contrôleurs. Il assure la gestion et le contrôle de cet ensemble. Partant d'un système totalement décentralisé, la prise de décision serait totalement basée sur des principes de coopération et de coordination. Afin de réduire les temps d'attente aux différents carrefours présents sur un axe et pour réduire le nombre d'arrêts des usagers, le CDZ est dédié à la coordination de cet axe. L'amélioration de tels critères de performance par le biais d'un phasage adapté a un impact direct sur la pollution et sur la consommation en carburant. Ces considérations poussent d'ailleurs à favoriser le passage des bus aux carrefours en utilisant des capteurs et des phasages adaptés.

Le Contrôleur (CTRL) ne possède qu'une connaissance très limitée du système car il ne gère qu'un nombre limité de carrefours à feux. Son rôle est d'assurer l'écoulement du trafic local. Il doit le faire en respectant le plan de feux fourni par le poste central ou le coordinateur de zone. Il peut cependant allonger ou raccourcir certaines phases de ce plan en réalisant de l'escamotage sur les phases. Il effectue une amélioration locale par rapport à une solution globale. Ceci est possible grâce à la définition même des plans de feux et de leurs marges de sécurité. Dans la majeure partie des systèmes existants, on trouve encore des carrefours non reliés au PC.

Une entité de régulation (un contrôleur, un coordinateur de zone ou le poste central) peut gérer un ou plusieurs applications parmi les quatre applications citées ci-dessus (CAF, MDT, VTC et PMV), mais dans chaque niveau de ces entités, les applications sont gérées différemment.

Afin de comprendre les fonctionnements et les objectifs des trois types d'entité de régulation, nous utilisons la méthode MaSE.

4- MaSE pour la conception du SSD

L'utilisation de MaSE pour conceptualiser le SSD des STU permet d'identifier et de définir les objectifs que le système doit réaliser et également de décrire le fonctionnement et les communications des entités afin de réaliser les objectifs du système. MaSE est composée de deux phases : une phase d'analyse et une phase de spécification.

4-1- Analyse

La phase d'analyse de MaSE produit un ensemble de rôles et de tâches, qui décrivent comment un système satisfait ses objectifs globaux. Les objectifs sont une abstraction des conditions détaillées et sont réalisés par des rôles. Typiquement, un système a un objectif global et un ensemble de sous-objectifs qui doivent être réalisés pour atteindre le but du système. On utilise des buts dans MaSE parce qu'ils expriment ce que le système essaye de réaliser pour être plus stable.

Un rôle décrit l'exécution d'une tâche par une entité dans le système, chaque rôle est responsable d'atteindre un objectif spécifique ou un sous-objectif.

4-1-1- Identifier les buts

La première étape dans la phase d'analyse est d'identifier les objectifs en transformant les spécifications initiales du système en objectifs structurés.

Le point de départ pour identifier les objectifs est le contexte initial du système et les spécifications et l'analyse des besoins. Ces besoins expriment les services que le système doit fournir et quel comportement le système doit avoir, selon ses données et son état actuel.

On peut citer quelques scénarios de problèmes rencontrés dans les systèmes du trafic urbain :

Premier scénario : On suppose que les voies d'entrée aux intersections critiques sont équipées par des capteurs et des détecteurs pour envoyer régulièrement des informations à propos de la situation actuelle du trafic à une entité de régulation. Grâce à ces données, l'entité de régulation et les décideurs doivent définir des actions pour minimiser les saturations par les moyens suivants :

- 1) Afficher des messages sur les panneaux à messages variables pour informer les usagers à propos des problèmes existants ou des accidents, et pour proposer des trajets alternatifs pour éviter les zones à problème.
- 2) Informer les décideurs et les centres de décisions pour maîtriser la situation ou pour envoyer les bonnes personnes capables de gérer ou résoudre le problème.

Comme l'infrastructure du contrôle de trafic devient de plus en plus complexe, il faut aider les décideurs à exécuter leurs tâches de gestion, et à configurer les plans de feux pour tout (ou partie) du réseau urbain afin de profiter au mieux des données disponibles, surtout dans les cas critiques comme le cas d'accident et les cas de saturation.

Deuxième scénario : dans les grandes villes, les bus sont équipés de moyens de communication : les conducteurs qui informent en permanence les décideurs de leurs identités et des localisations des bus. L'objectif essentiel de la gestion du transport en commun est de détecter les incidents de bus : panne, absence, avance ou retard par rapport au planning, ... etc. de manière à réagir à ces incidents pour rétablir une qualité de service acceptable.

Plusieurs méthodes sont possibles pour améliorer la qualité de service des véhicules de transport en commun :

- 1) Appliquer des stratégies pour donner la priorité aux VTC sur les carrefours à feux (voir annexe C).
- 2) Garder une communication continue avec le conducteur pour lui demander d'accélérer ou de ralentir dans l'objectif de respecter son planning.

L'identification des objectifs se fait en deux étapes :

- Etape 1 : Numéroté les objectifs

Dans cette étape il faut faire sortir l'essentiel à partir des besoins, extraire des scénarios à partir des spécifications initiales et décrire l'objectif de chaque scénario.

Nous pouvons identifier les objectifs suivants :

- 1- Optimiser la circulation « *QUOI* »
- 2- Identifier les zones critiques
- 3- Identifier les intersections les plus importantes (par rapport au nombre de véhicules servis)
- 4- Calculer un nouveau plan de feux
- 5- Calculer la charge pour chaque axe dans les intersections les plus importantes (« *COMMENT* » : débit, concentration- densité, taux d'occupation)
- 6- Appliquer un message sur un panneau à message variable

- 7- Suivre un véhicule de transport en commun
- 8- Donner la priorité à un véhicule de transport en commun
- 9- Détecter un accident, une saturation, une panne (informer le système externe).
- 10- Proposer une alternative, une solution.
- 11- Expliquer un événement
- 12- Evaluer une modification proposée
- 13- Coordonner un plan de feux entre les intersections adjacentes (appliquer une onde vert)
- 14- Résoudre un conflit entre plusieurs messages proposés (PMV)
- 15- Recueillir les données depuis les capteurs
- 16- Communiquer

- Etape 2 : Structurer les objectifs

Organiser les objectifs d'une façon hiérarchique dans un graphe acyclique où les nœuds représentent les objectifs et les arcs définissent la relation entre un objectif et ses sous-objectifs, ce graphe n'est pas un arbre puisque un objectif peut être un sous-objectif de plus d'un objectif père.

Les objectifs sont organisés selon leurs importances et les relations entre eux, les sous-objectifs sont généralement plus faciles à gérer et à comprendre.

La première étape pour construire un diagramme hiérarchique des objectifs est d'identifier l'objectif global du système qui sera placé tout en haut dans le diagramme.

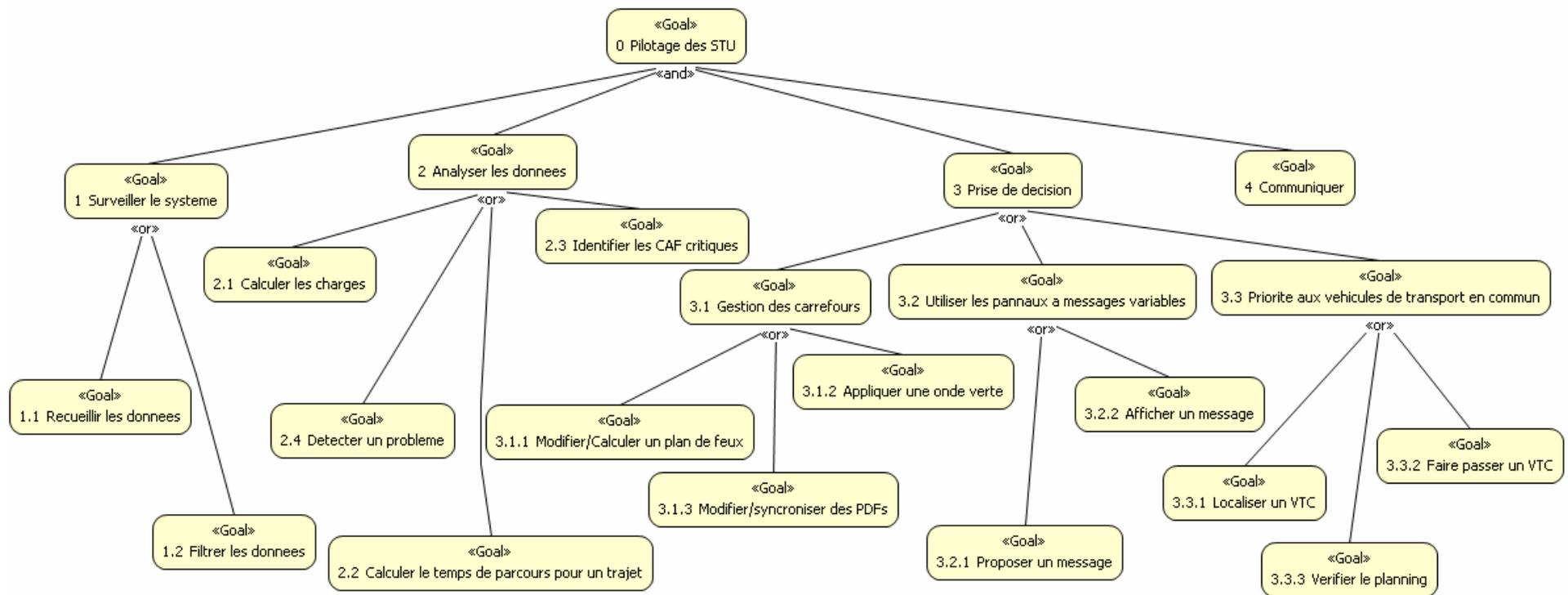


Figure 4-8 : Structure hiérarchique des objectifs.

Les services techniques des STU ont pour objectif d'offrir de bonnes conditions de circulation aux usagers, de maîtriser les conditions de saturation quelque soit l'état du système et d'optimiser la circulation, soit en modifiant la conception de l'infrastructure soit en modifiant le pilotage et la réglementation. La figure 4-8 montre le modèle d'objectif du pilotage des STU. L'objectif global "*Pilotage des STU*" est décomposé en sous-objectifs en utilisant les raffinements AND/OR. Il est composé de trois sous-objectifs : "*Surveiller le système*", "*Analyser les données*" et "*Prise de décision*". Ensuite chacun de ses deux objectif est composé d'un ou de plusieurs sous-objectifs:

+ "*Surveiller le système*" doit d'abord "*Recueillir les données*" puis "*Filtrer les données*" selon les besoins.

+ "*Analyser les données*" a comme sous-objectifs de calculer les charges sur les courants d'entrée aux carrefours, d'identifier les zones et les carrefours critiques, de détecter l'existence des cas anormaux (accident, panne, saturation, ... etc.) et d'obtenir certains statistiques qui aident à calculer le temps de parcours d'un trajet.

+ "*Prise de décision*" peut être divisé en trois catégories : "*Gestion des carrefours*", "*Utilisations des panneaux à messages variables*" et "*gestion des véhicules de transport en commun*".

L'objectif "*Gestion des carrefours*" comprend "*modifier/calculer des plans de feux*", "*synchroniser les plans de feux sur les carrefours adjacentes*" et appliquer certaines méthodes d'optimisation comme l'onde verte.

L'"*Utilisation des panneaux à messages variables*" a pour objectif de montrer aux usagers des informations utiles concernant l'état actuel du système, i.e. l'existence d'une panne ou d'un cas de saturation quelque part dans le système; montrer le temps approximatif nécessaire pour atteindre un lieu précis; ou proposer un trajet alternatif afin d'éviter les congestions.

"*Gestion des véhicules de transport en commun*" a pour objectif de localiser tous les véhicules de transport en commun, de vérifier s'ils respectent leurs plannings, et de garder un contact continu avec les conducteurs afin de pouvoir leur demander d'accélérer ou de ralentir selon la situation par rapport au planning. Il est possible également d'appliquer certaines méthodes ou stratégies pour donner la priorité de passage à ces véhicules sur les carrefours à feux afin d'améliorer leur performance.

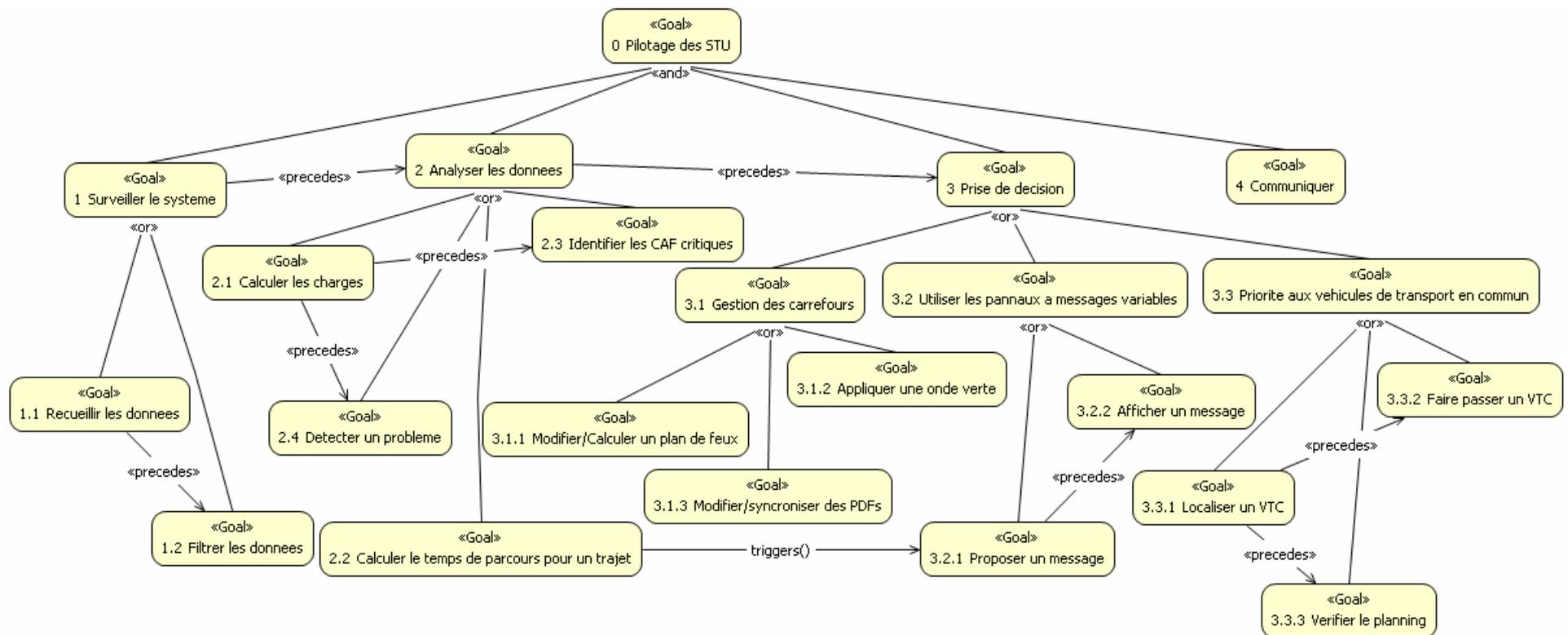


Figure 4-9 : Structure hiérarchique des objectifs (modèle dynamique)

Pour raffiner ce modèle d'objectifs et pour présenter le dynamisme du système, et montrer la séquence de réalisation des objectifs, d'autres notations sont ajoutées au modèle précédant : *precedes*, *trigger()* comme le montre la figure 4-9.

Par exemple, le processus de pilotage commence par une phase de surveillance "*Surveiller le système*", suivie par une phase d'analyse, et à la fin l'étape de prise de décision.

Surveiller → Analyser → Décider

4-1-2- Le modèle de rôles

Cette étape est essentielle pour convertir les objectifs en rôles et tâches associés. A partir des besoins, il faut extraire et décrire la séquence des événements définissant le comportement désiré du système. Cette étape est aussi nécessaire pour déterminer les communications entre les entités.

Le modèle de rôles est extrait à partir du contexte initial du système, cela doit inclure le comportement normal du système (les cas positifs) et le comportement anomal : accident, panne, ...etc. (les cas négatifs). Il est normal que les diagrammes de *cas d'utilisation* ne puissent pas décrire toutes les possibilités du système mais ils aident à décrire le chemin de communication et les rôles.

- Raffiner les rôles :

Le but est de transformer le diagramme hiérarchique d'objectifs en rôles et associer des tâches à chaque rôle. Les rôles sont le noyau des classes d'agents, et ils correspondent aux objectifs du système. Dans cette étape chaque objectif doit être associé à un rôle et chaque rôle doit être joué par un agent.

Normalement chaque objectif est associé à un seul rôle mais il est possible qu'un rôle soit responsable de plusieurs objectifs, les objectifs qui sont en relation peuvent être combinés dans un seul rôle (dans ce cas le rôle devient plus complexe mais cela simplifie la conception globale).

L'interaction du système avec des ressources externes ou internes exige un rôle spécifique comme une interface pour le reste du système. Le décideur peut être

peut correspondre à un carrefour à feux, ou l'application d'une méthode de calcul des nouveaux plans de feux.

Le rôle "VTC" a comme objectif de gérer le fonctionnement des véhicules de transport en commun, et le rôle "PMV" gère les panneaux à messages variables. Le rôle RUI est le rôle d'interface utilisateur, il sert d'intermédiaire entre le décideur et le système.

Le modèle de rôle de la figure 4-10 peut être représenté d'une manière plus simple, comme le montre la figure 4-11 :

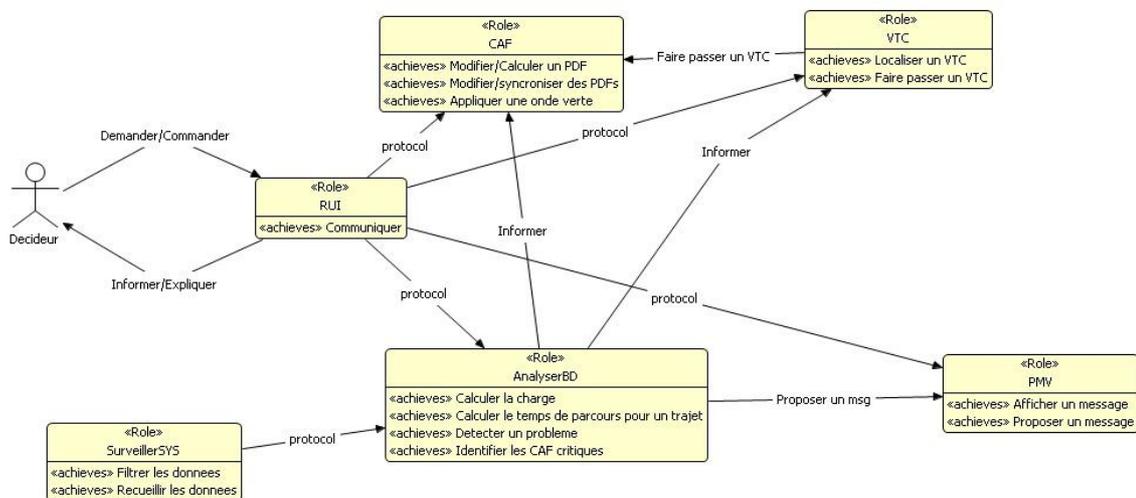


Figure 4-11 : Le modèle de rôle (simplifié)

Une tâche est réalisée par un et un seul rôle. Le comportement de chaque rôle peut être décrit en détaillant ses tâches et les communications entre rôles.

Dans la phase d'analyse, nous avons construit le modèle d'objectifs et le modèle de rôle. L'étape suivante est de transformer ces modèles en modèles de conception qui correspondent mieux à l'implémentation.

4-2- Conception

Dans cette phase, le modèle de rôle est transformé en modèle de classes d'agents, et les interactions présentes dans le modèle de rôle seront transformées en interactions entre les agents qui jouent ces rôles. Ces interactions seront modélisées en utilisant les diagrammes de protocoles (diagrammes de séquence AUML). Les détails des agents sont présentés dans un modèle de plan pour chaque agent.

Dans le contexte du SSD, les agents essentiels représentent les entités de régulation (poste central, coordinateur de zone et contrôleur).

4-2-1- Construire le modèle de classes d'agents

La création de modèles de classes d'agents est faite en associant les rôles, prédéfinis dans l'étape d'analyse, à leurs agents. Le diagramme de classe d'agent donne une vue globale de l'organisation du système et les communications entre ses parties. Chaque agent est défini par les rôles qu'il va jouer et par les communications auxquelles il va participer.

La figure 4-12 montre un modèle général d'agents, nous y trouvons l'agent *ER* (entité de régulation) qui joue les rôles *AnalyserBD*, *CAF*, *VTC* et *PMV*, et l'agent *BD* (base de données) jouant le rôle *SurveillerSYS*, et l'agent *IU* (interface utilisateur) jouant le rôle d'intermédiaire entre le system et le décideur.

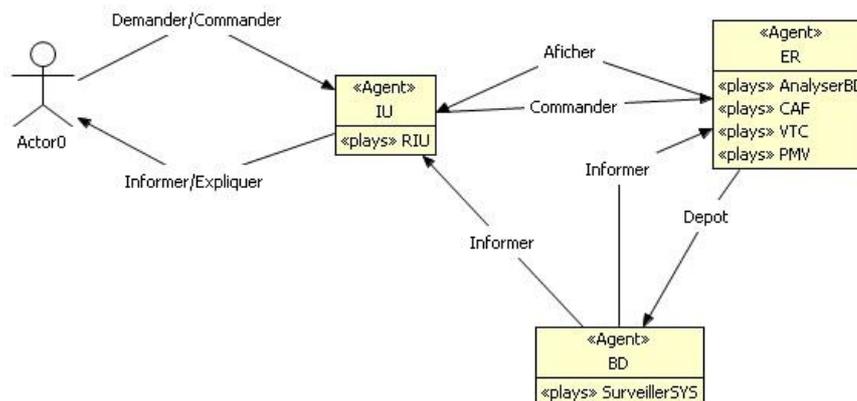


Figure 4-12 : modèle général d'agents

Comme nous avons montré précédemment qu'une entité de régulation (ER) peut être un poste central (PC), un coordinateur de zone (CDZ) ou un contrôleur (CTRL). Chacun de ces trois types d'entité est représenté par un agent. Nous obtenons par conséquent le schéma de la figure 4-13. Les trois agents *PC*, *CDZ* et *CTRL* existent en trois niveaux hiérarchiques, chaque agent commande les agents du niveau inférieur (La relation *Commander*), et fournir l'agent du niveau supérieur (La relation *informer*). Le *décideur* peut *commander* le système ou lui *demander* certaines informations, et le système peut *informer* le décideur ou lui *expliquer* certaines situations.

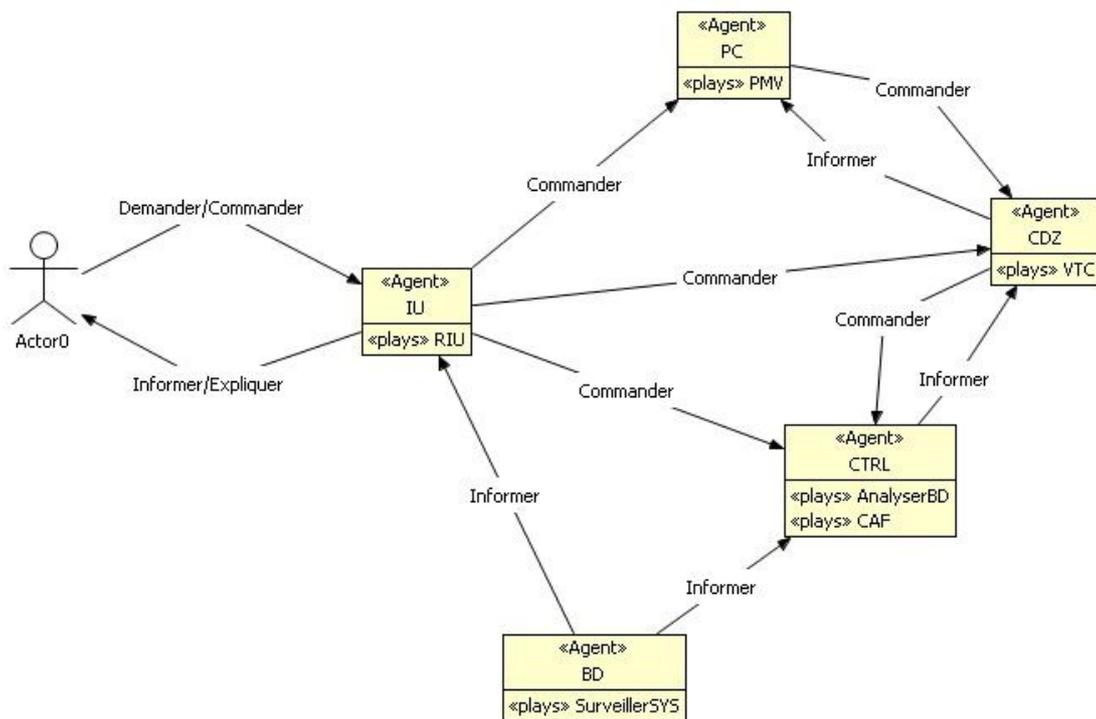


Figure 4-13 : modèle d'agents

Les rôles sont le noyau de conception du SMA, ils sont utilisés pour définir les objectifs du système.

Le modèle de rôle capture les fonctionnements basics du système, les rôles joués par chaque agent sont encapsulés dans le corps de la classe d'agent, chaque rôle est précédé par le mot « *play* », et l'agent est noté par le mot « *agent* ».

Les interactions entre les rôles dans le modèle de rôle correspondent aux interactions entre les agents appropriés.

Une fois le modèle d'agent construit, les classes d'agents et les protocoles sont identifiés mais ne sont pas encore définis, les modèles de protocoles et les modèles de plans sont nécessaires pour conceptualiser les agents individus.

4-2-2- Construire les interactions et les communications

Dans le diagramme de classe d'agents les communications entre agents sont identifiées mais le but de cette étape est de les détailler grâce à leurs tâches concurrentes.

L'objectif du modèle de protocole est de définir les détails des protocoles identifiés dans les modèles de rôles et les modèles de classes d'agents, le modèle de

protocole défini les protocoles en termes de messages envoyés entre les agents, ou entre les agents et les acteurs externes.

Les diagrammes de protocole pour les SMA sont similaires aux diagrammes standards de séquence d'UML mais ici ils sont utilisés pour montrer la séquence d'événements entre les rôles et pour définir les communications entre les agents qui vont jouer les rôles associés.

L'initiateur commence la conversation en envoyant le premier message, quand le récepteur reçoit le message il va le comparer avec les conversations actuelles, s'il trouve une tâche associée qui correspond à ce message il passe à un autre état et il commence à exécuter le processus de raisonnement de cette tâche.

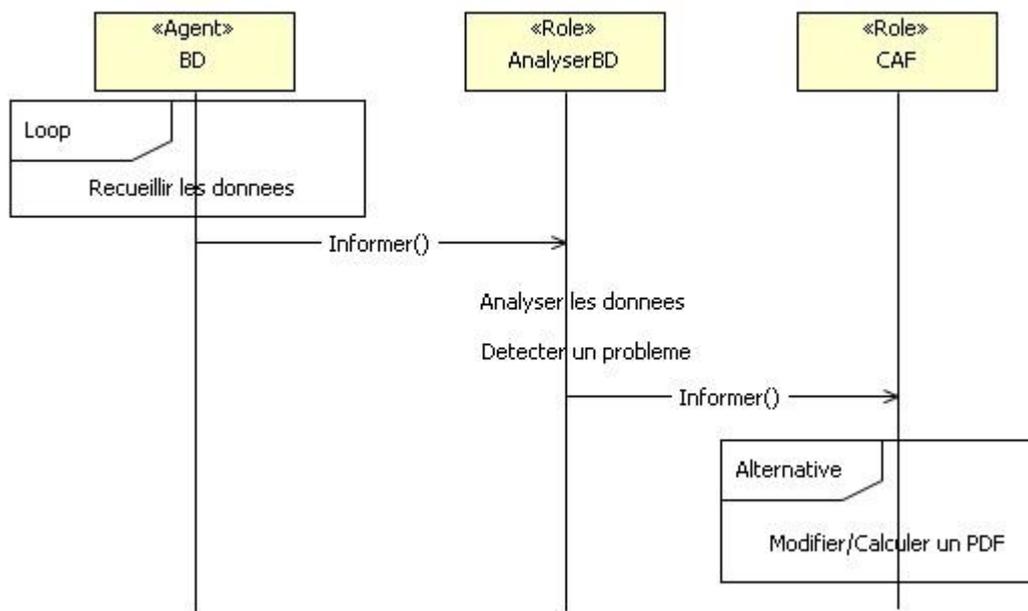


Figure 4-14 : protocole1

Protocole 1

L'agent *contrôleur* demande à l'agent BD les données qui le concernent, l'agent BD va filtrer les données selon la demande et il les envoie au contrôleur. L'agent contrôleur commence à analyser les données reçues, une fois le problème détecté, une phase de prise de décision commence. Une fois la décision prise, la base de données sera enrichie et le contrôleur informe son supérieur et le décideur de la décision.

Protocole 2

Les interactions entre le décideur et le système multi-agents se font comme des dialogues, le décideur peut commander au système exécuter certaines tâches, comme le changement de phasage, ou bien il peut demander au système d'étudier les conséquences résultant d'une action quelconque. Le système à son tour exécute l'action requise, il analyse ses résultats et il en informe le décideur. Le SMA doit aussi surveiller le système étudié régulièrement et informer le décideur par les événements rencontrés. L'objectif du rôle d'analyse dans chaque entité de régulation essaie de comprendre les causes et les conséquences des événements.

4-2-3- Les modèle de plan d'agents

La dernière étape dans la phase de conception est la construction de l'architecture interne des agents et leurs processus de raisonnement en utilisant un modèle de plan pour chaque agent. Un plan représente les moyens par lesquels un agent réalise un objectif. Ce plan peut être un algorithme.

Il faut d'abord définir l'architecture des agents et ensuite définir la structure de ses composantes. Chaque agent est présenté par au moins un modèle de plan.

Le modèle de plan de l'agent contrôleur est similaire à un diagramme d'état transition, ce processus commence quand le contrôleur s'informe auprès de l'agent BD pour obtenir les données qui le concernent. Après avoir reçu ces données, il les analyse, s'il trouve un cas anormal, il entre dans une phase de prise de décision pour résoudre ce problème. Une fois la décision prise, il l'applique et il en informe son supérieur et le décideur.

Un plan représente les moyens par lesquels un agent peut satisfaire un objectif dans son organisation, et peut être vu comme un algorithme pour atteindre un objectif spécifique.

Le nombre de plans de chaque agent dépend de la structure de l'agent, normalement, il faut un plan pour chaque rôle, ou plusieurs si l'agent a le choix entre plusieurs plans pour atteindre le même objectif.

Dans le modèle de plan, les transitions entre les différents états de l'agent se passent suite à l'envoi ou la réception des messages des autres agents ou acteurs. Les primitives suivantes sont utilisées pour les transitions :

[guard] receive(message, sender)

send (message, receiver)
message = performative(p1, ..., pn)

Chaque état a un ensemble d'actions à exécuter successivement. Une action est une fonction qui retourne une valeur. Un agent passe d'un état à l'autre suite à la réception des valeurs de toutes les actions.

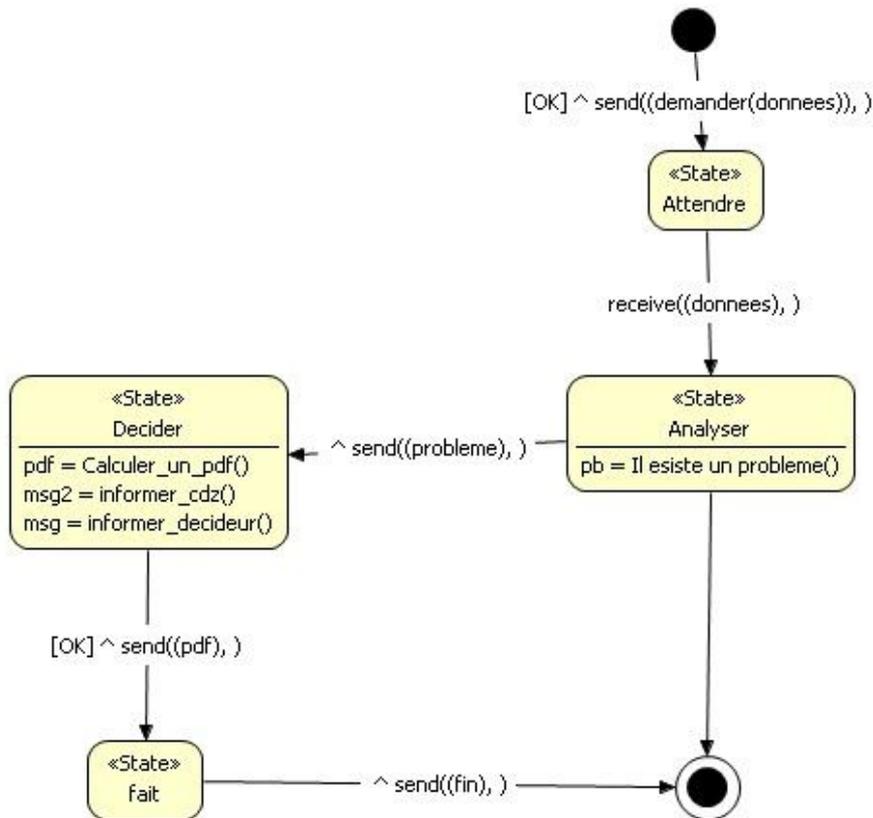


Figure 4-15 : modèle de plan de l'agent entité de régulation

5- Conclusion

Nous avons présenté une démarche de conception pour le sous-système décisionnel des STU en utilisant MaSE et son outil AgentTool. Cette approche est simple et en même temps assez détaillée. Cette démarche a permis de conceptualiser le sous-système décisionnel des STU, les objectifs sont identifiés, le modèle de rôles et le modèle d'agent sont construits. Le comportement de chaque agent est détaillé. Nous avons montré le passage d'une tâche à l'autre afin d'obtenir une bibliothèque riche des modèles suivants : le modèle d'objectif, le modèle de rôle, le modèle de classe d'agents, les modèles de protocole et les modèles de plan d'agent. Les derniers

trois modèles correspondent à la conception du SSD et ils nous permettent de passer facilement à la phase d'implémentation.

Les entités de régulation (Contrôleur, Coordinateur de zone et Poste Central) sont représentées par des agents. L'implémentation de ce système sera présentée dans le chapitre suivant, Nous utilisons Java comme langage de programmation et Madkit comme plateforme d'implémentation, les méta-modèles et un modèle organisationnel AGR proposés par MadKit facilitent l'implémentation de notre modèle.

Chapitre 5

-Conception et Implémentation : Une Architecture globale d'un système multi-agents pour le SSD

Sommaire

1- INTRODUCTION	111
2- FORMALISER LA CONNAISSANCE DU MODÈLE	112
3- LE MODÈLE MULTI-AGENTS :	116
<i>3-1- Madkit et son modèle d'organisation Alaadin-AGR.....</i>	<i>116</i>
<i>3-2- Passage de la conception d'agent à leur implémentation :.....</i>	<i>117</i>
4-MODÈLE DE RÉSULTATS.....	125
5- CONCLUSION	126

1- Introduction

Ce chapitre est dédié à la réalisation de notre système multi-agents. Les modèles obtenus dans le chapitre précédent seront traduits dans des composants logiciels. Nous avons choisi MadKit comme plateforme d'implémentation. Le fait que cette plateforme soit basée sur le paradigme AGR et son langage de programmation étaient les critères principaux motivant ce choix. (L'implémentation de notre système multi-agents est indépendante de sa conception).

La réalisation du modèle du SSD est structurée autour du modèle MVC (Modèle-Vue-Contrôleur) (Crawford et Kaplan, 2003). Elle s'intègre dans les travaux présentés dans (Hostin, 2006). Le modèle MVC est une architecture et une méthode de conception qui organise l'interface Homme-machine d'une application logicielle. Il divise cette interface en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface. La figure 5-1 montre cette architecture.

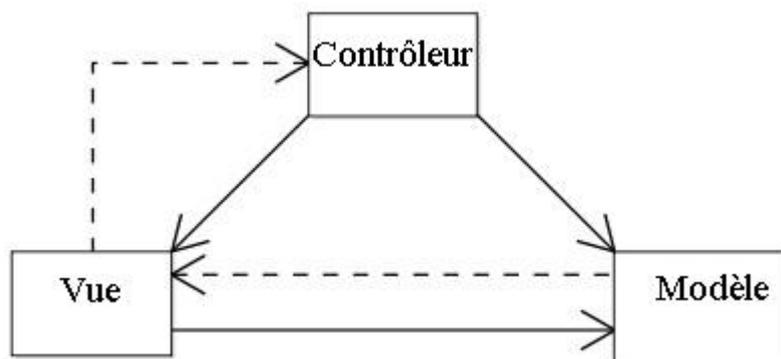


Figure 5-1 : Le Modèle MVC et les différentes interactions entre ses modules.

La figure 5-2 montre l'architecture d'implémentation grâce à ASCImi-STU. Nous présentons dans les paragraphes suivants les éléments de cette architecture :

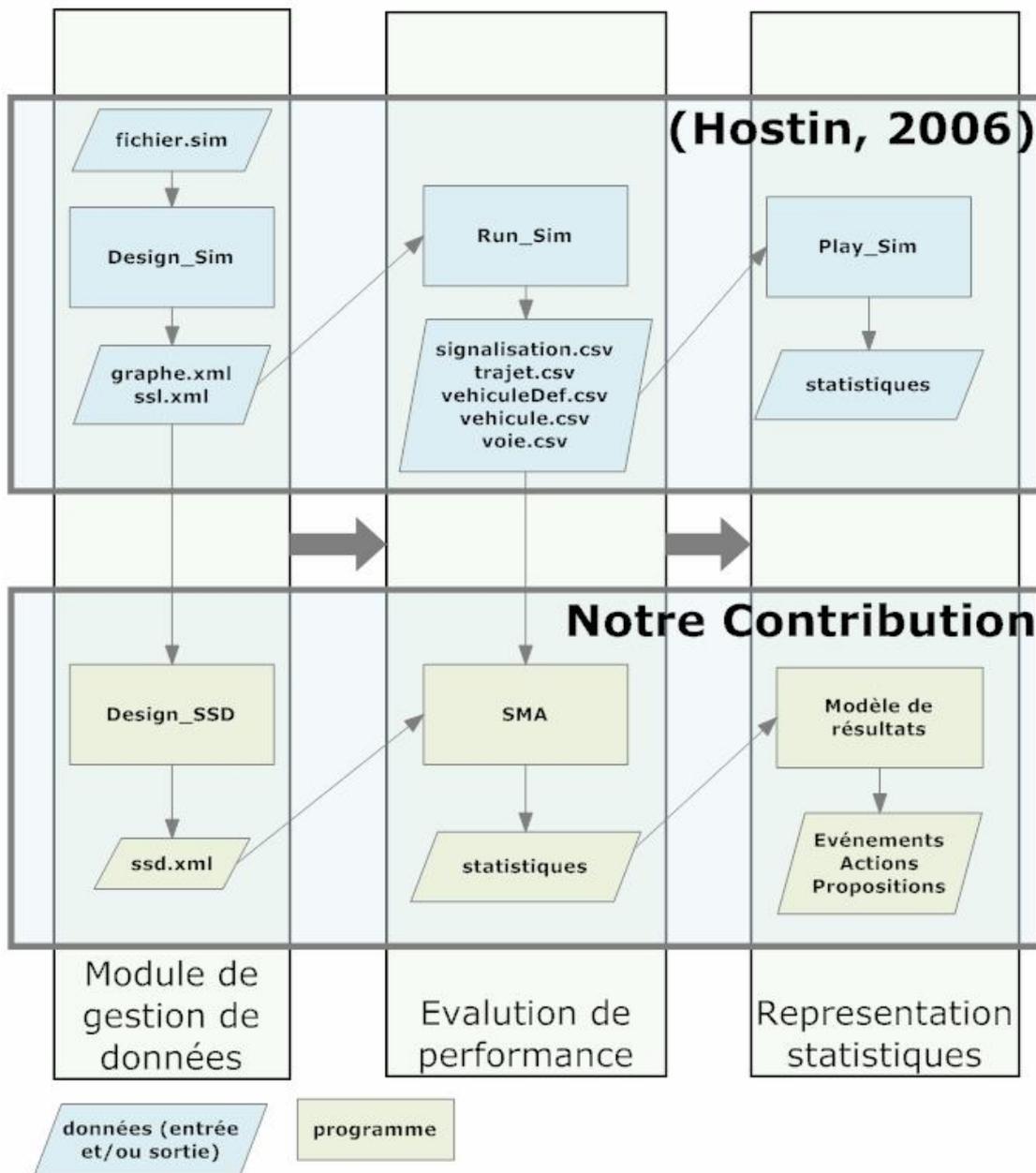


Figure 5-2 : architecture d'implémentation.

2- Formaliser la connaissance du modèle

Selon la méthodologie ASCI-mi STU proposée par (Sarramia, 2002), il est possible de modéliser le sous-système physique au niveau microscopique ou macroscopique ou combiné. La description microscopique d'un composant macroscopique est un graphe microscopique, i.e. un réseau est composé de deux parties : la première partie est un graphe macroscopique comprenant des arcs et des nœuds macroscopiques (macro), et la deuxième partie est un ensemble de graphes microscopiques, chaque élément de cet ensemble contient des arcs et des nœuds microscopiques (micro).

(Hostin, 2006) a ajouté une interface graphique permettant de décrire les deux sous-systèmes : physique et logique d'un réseau aux deux niveaux microscopique et macroscopique, les fichiers de sorties sont des fichiers type XML (voir figure 5-2):

- *graphe.xml* pour la description du réseau,
- *ssl.xml* pour la description du sous système logique.

La figure (5-3) montre le modèle physique et logique d'un carrefour à feux. Au niveau macroscopique ce carrefour comprend un nœud macro et quatre arcs macro, au niveau microscopique, chaque élément macro comprend un ensemble d'arcs micro (les flèches) et de nœuds (les points).

Cependant, cette description ne contient pas les entités décisionnelles. Du point de vue décisionnel, un carrefour est considéré comme une ressource commune à plusieurs arcs microscopiques, l'ensemble des arcs micro qui utilise le même plan de feux sur un carrefour forme une ligne de feux, une ligne de feux est composée de un ou plusieurs courants, un courant est un ensemble d'arcs micro entrant dans le carrefour par la même entrée (selon la figure : les arcs micro 1 et 2 forment un courant : cour1, et les arcs 3 et 4 forment un deuxième courant cour2. les deux courant cour1 et cour2 appartiennent à la même ligne de feux).

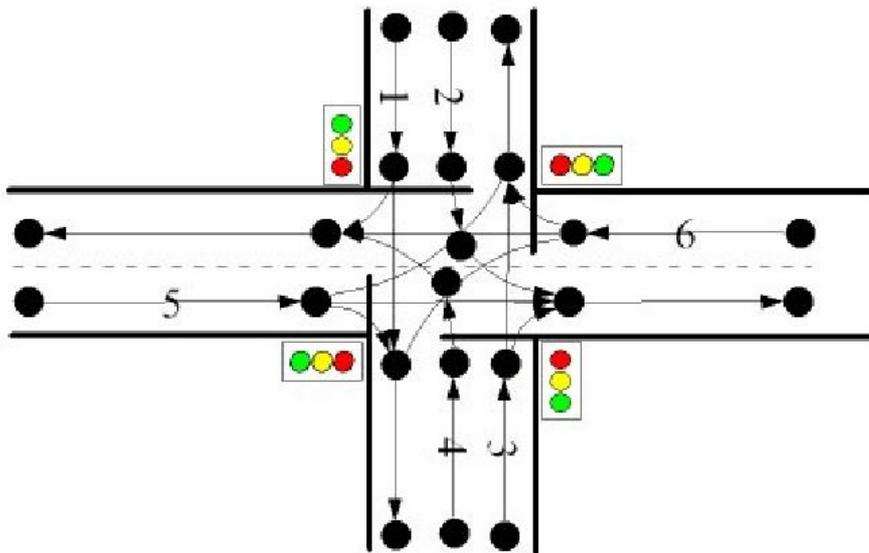


Figure 5-3 : Diagramme d'un carrefour

Le tableau 5-1 résume la différence entre les deux descriptions :

Description physique et logique (<i>graphe.xml, ssl.xml</i>)	Description décisionnelle (<i>ssd.xml</i>)
Graphe macro Nœuds_macro Arcs_macro Ensemble de graphes micro Nœuds_micro Arcs_micro	Poste Central Coordinateurs de zones Contrôleurs Carrefour à feux Ligne de feux Courants Arcs_micro

Tableau 5-1 : description décisionnelle vs description physique et logique

Pour passer de la description physique et logique à la description décisionnelle on peut utiliser les deux fichiers *graphe.xml* et *ssl.xml* pour extraire la description décisionnelle des carrefours à feux, l'algorithme de la figure 5-4 montre ce processus :

<p>Première partie : <i>Pour tout nœud micro type FeuMicro dans graphe.xml</i> <i>Obtenir le plan de feux utilisé</i> <i>Obtenir l'arc micro entrant</i> <i>Obtenir l'arc macro contenant le nœud micro</i> <i>Obtenir la destination de l'arc macro (un nœud macro)</i></p> <p>Deuxième partie : <i>La liste de carrefour à feux = la liste de nœud macro obtenus dans la première partie</i> <i>Pour chaque carrefour :</i> <i>Une ligne de feux correspond à chaque plan de feux</i> <i>Pour chaque ligne de feux :</i> <i>Ajouter un courant pour chaque couple (arc macro \leftrightarrow plan de feux)</i> <i>Pour chaque courant</i> <i>Ajouter la liste des arcs micro entrant à ses nœuds micro type FeuMicro</i></p>

Figure 5-4 : Extraction quasi-automatisation du *ssd.xml* à partir de *graphe.xml* et *ssl.xml*

Un extrait du contenu du fichier obtenu par cet algorithme est présenté en Figure 5-5 :

```

- <CAF Name="N100">
- <LDF Name="N100-RN9">
- <Courant Name="A2">
  <ArcMicro Name="A2-a2" Target="A2-n4" />
  <ArcMicro Name="A2-a3" Target="A2-n5" />
</Courant>
- <Courant Name="A103">
  <ArcMicro Name="A103-a1" Target="A103-n2" />
</Courant>
- <Courant Name="A106">
  <ArcMicro Name="A106-a2" Target="A106-n3" />
  <ArcMicro Name="A106-a3" Target="A106-n5" />
</Courant>
</LDF>
- <LDF Name="N100-Zola">
- <Courant Name="A102">
  <ArcMicro Name="A102-a1" Target="A102-n2" />
</Courant>
- <Courant Name="A104">
  <ArcMicro Name="A104-a2" Target="A104-n3" />
</Courant>
</LDF>
</CAF>

```

Figure 5-5 : Exemple de contenu quasi-automatisé du fichier ssd.xml

A ce point, le fichier ssd.xml obtenu n'est pas complet puisque les entités de régulation, type : contrôleurs, coordinateurs de zones et le poste central, ne sont pas encore pris en compte. Une interface graphique permet à l'utilisateur de définir la hiérarchie de cette structure, et faire appartenir les carrefours à feux déjà obtenus à leurs propres contrôleurs. La figure 5-6 montre une capture de l'écran de cette interface :

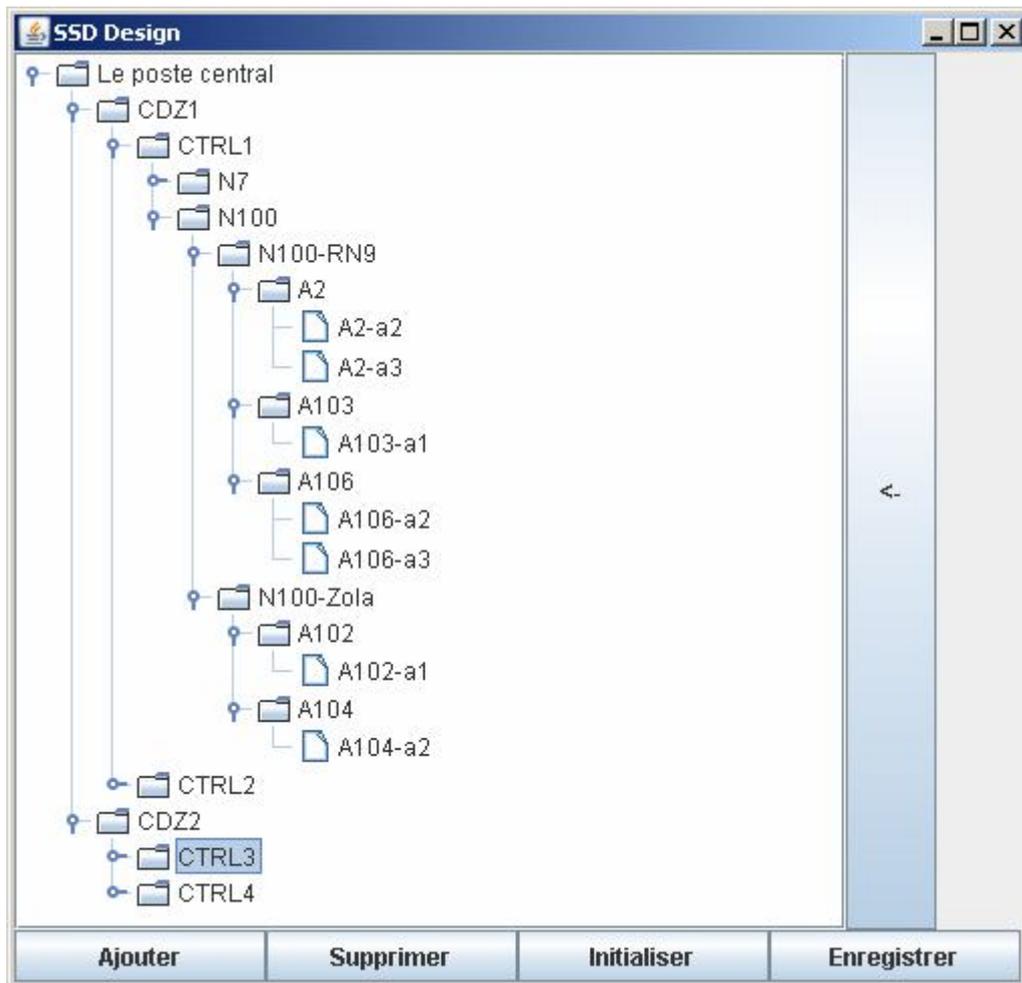


Figure 5-6 : Capture d'écran du module : SSD_Design

3- Le modèle multi-agents :

3-1- Madkit et son modèle d'organisation Alaadin-AGR

Pour passer du modèle de conception au modèle d'implémentation nous avons choisi, comme environnement d'implantation des SMA, la plateforme MadKit (Gutknecht O., Ferber J., 1999). Elle implémente le paradigme Agent-Group-Rôle (AGR) présenté précédemment dans le chapitre 3 (3-4-3). La figure 5-7 montre une représentation simple de ce paradigme. C'est une organisation vue comme un cadre pour les activités et les interactions des agents en définissant des groupes, des rôles et des interconnexions entre ces agents.

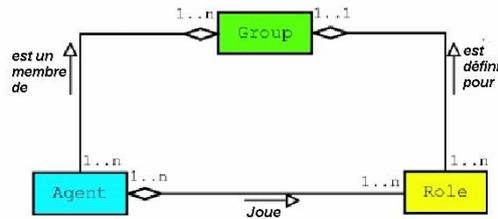


Figure 5-7 : L'architecture Agent, Group, Rôle (AGR)

La classe de base d'un agent Madkit (AbstractAgent) définit quelques fonctionnalités de base pouvant être nécessaires dans les modèles classiques, ces fonctions sont :

- Cycle de vie : l'agent dispose de quatre états (création, activation, exécution et destruction, Figure 5-8) et la possibilité de démarrer d'autres agents sur le noyau local. Aucun mécanisme d'exécution n'est défini à ce niveau.

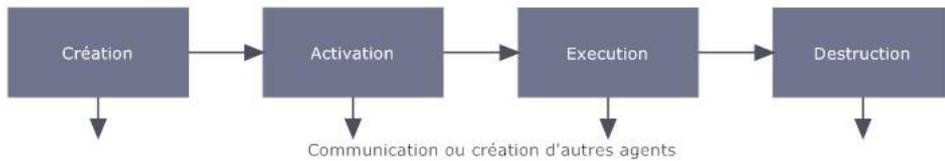


Figure 5-8 : Cycle de vie d'un agent sous madkit.

- Communication : elle est implémentée sous forme de passage de messages asynchrones. (la communication exige un support physique, un langage et un mode de diffusion. La diffusion est anonyme (tableau noir ou envoi multiple) ou nominatif)
- Organisation : tout agent dispose de primitives permettant de connaître les groupes et les rôles courants.
- Outils : la classe de base des agents permet de manipuler une interface graphique associée à l'agent.

3-2- Passage de la conception d'agent à leur implémentation :

Selon les équipementiers de régulation du trafic urbain, l'orientation actuelle du pilotage des STU est de rendre chaque organe de régulation (i.e. agent) le plus autonome possible, surtout au niveau décisionnel. Cependant, si celui-ci n'est pas capable de résoudre son problème, il doit coopérer avec d'autres organes de régulation. Nous avons doté chaque agent de nombreuses méthodes de micro et de macro régulation telles que spécifiées par le CERTU.

Selon l'approche retenue pour étudier un système, la modélisation nécessite un ou plusieurs niveaux de détail macroscopique, mésoscopique ou microscopique.

Un agent réagit par rapport aux événements qu'il perçoit de son environnement, déclencheurs de ses différents processus. Ces événements sont soit des décisions provenant d'autres agents du SSD, soit un changement d'état d'un objet du SSP (dépassement d'un seuil de concentration, panne d'un feu, etc.). Ces événements extérieurs entraînent des événements internes au fonctionnement de l'agent.

Un agent possède deux états : il est au repos, lorsqu'il est en attente d'un événement déclenchant; il est actif, lorsqu'un événement déclenchant survient (réveil par le PC).

Il procède alors à ses différentes activités. Dans ce cas, il utilise un algorithme de résolution pour effectuer une des activités suivantes :

- La récolte des données relatives à son environnement, par rapport à l'événement qui l'a réveillé;
- L'analyse des données permet de décider si son action est nécessaire ou non selon des critères fixés. Ceci permettra de déterminer s'il s'agit d'une action réflexe ou non;
- L'activité d'action décidée suite à l'analyse des données, correspond, par exemple au calcul d'un nouveau plan de feux. Les résultats ainsi obtenus doivent être transmis à d'autres agents, et/ou aux entités contrôlées par cet agent;
- L'activité de communication offre à l'agent la possibilité d'informer ou de solliciter les autres agents du SMA.

Les cinq agents : Interface Utilisateur (IU), Base de données (BD), CTRL, CDZ et PC seront implémentés en Java dont la classe mère est la classe « Agent » fournie par Madkit (figure 5-9) :

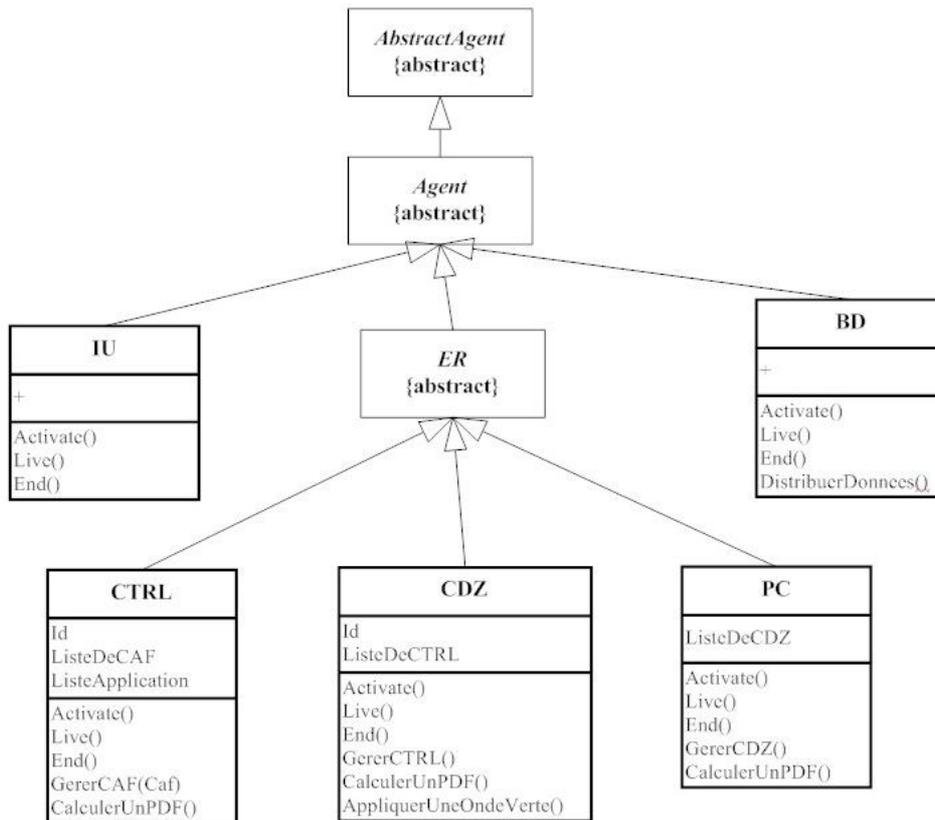


Figure 5-9 : Diagramme de classe d'agents.

MadKit permet d'organiser les entités décisionnelles selon son architecture AGR comme suit :

- La topologie du réseau du STU étudié est composée de plusieurs zones géographiques, chaque zone dirigée par un coordinateur de zone ou un contrôleur correspond à un groupe (figure 5-10). Un groupe spécial est celui du poste central : il permet, dans un premier temps, de décrire le système étudié (initialisation des données des différents agents), puis de les activer.

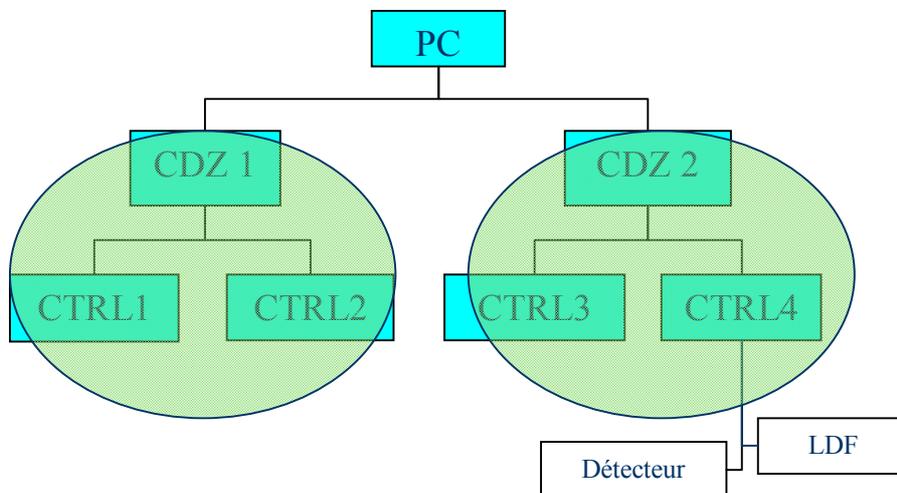


Figure 5-10 : Hiérarchie des entités décisionnelles du SSD.

Déroulement du SMA :

Le système multi-agents implémenté prend en entrée les fichiers de sortie du module Run_Sim (voir figure 5-2), les fichiers *vehicule* et *voie* sont les deux fichiers les plus importants puisqu'ils contiennent les données dynamiques :

- *Vehicule* : Contient la position, la vitesse et l'identifiant de l'arc sur lequel chaque voiture se déplace.
- *Voie* : Contient les informations concernant les voies (les arcs micros) comme le nombre de véhicules présents, entrées ou sortis, le nombre de véhicules en attente sur un parking, en fonction du temps.

Chaque agent CTRL surveille ce qui se passe dans la partie du système dont il est responsable et il réagit par rapport aux événements rencontrés, d'une manière autonome mais locale, et il communique périodiquement avec son supérieur (son propre CDZ) pour lui passer des informations agrégées concernant les carrefours gérés par lui, et les courants d'entrée à ces carrefours, il envoie également les événements et les décisions prises.

L'agent CDZ ayant une vision plus globale de sa zone, analyse les données et les événements envoyés par les agents CTRLs pour prendre des actions plus matures par rapport aux événements. A son tour il communique avec l'agent PC pour lui passer des statistiques et des informations détaillées sur les événements et les actions.

L'agent PC qui joue le rôle d'ordonnancement (ce rôle a été ajouté à l'agent PC au niveau d'implémentation mais il n'est pas nécessaire au niveau de conception). Le PC, dans sa phase d'activation, initialise la connaissance du système étudié grâce à son fichiers d'entrée « *ssd.xml* », il crée le groupe général auquel il appartient, et les autres groupes correspondants aux zones géographiques, il crée ensuite les autres agents : l'agent BD, les CDZs et les CTRLs. Lorsque tous les agents sont prêts, l'agent PC commande l'agent BD pour distribuer à chaque CTRL les données qui le concernent pour une période donnée, voir l'algorithme de la figure 5-11 :

Lecteur des fichiers d'entrée Voie et véhicule

*Création de la liste des courants (fichiers *ssd.xml*)*

Calculer la charge pour chaque courant :

*Pour la période d'étude et pour chaque instant *t**

Pour chaque entré dans voie

Identifier le courant contenant la voie

*Augmenter la charge de ce courant au moment *t* par 1*

Fin de calcul de la charge des courants

Associer les charges des courants à leurs LDFs → CAFs → CTRL

Figure 5-11 : Algorithme de distribution de données (calcul de charge par courant).

Les CTRLs commencent leur phase d'exécution pour analyser, détecter l'occurrence des événements anormaux et prendre les actions nécessaires. Ils se retrouvent sur un rendez-vous avec les CDZs et le PC à la fin de la période. La figure 5-12 montre l'algorithme de surveillance (détection de saturation).

Définir Seuil, Durée critique

*Parcourir le fichier de charge pour chaque courant (instant *t*, charge) :*

Si charge > seuil

Si nouvelle détection

**durée* = 1*

Sinon

**durée*++*

*Garder l'instant *t'* de saturation*

Sinon

Si fin de saturation

*Si *durée* > *durée critique* alors *saturation* = [*t'*, *durée*]*

Commencer une nouvelle détection

Figure 5-12 : Algorithme de détection de saturation (Surveillance).

Suite à la détection de saturation, le contrôleur responsable de la section saturée doit réagir et proposer un nouveau plan de feux. Il est facile de calculer un nouveau plan de feux au niveau du carrefour (algorithme de la figure 5-13):

Calculer les charges sur les courants

Pour chaque ligne de feux

Déterminer le courant dominant

Calculer la nouvelle durée du vert pour chaque courant par rapport à la charge

$$Vert_n \sim = Charge_n * durée\ totale\ du\ vert / Sum(Charge)$$

Calculer les nouveaux PDF pour le carrefour à partir des durées calculées et la matrice de sécurité.

Figure 5-13 : Algorithme de calcul du nouveau plan de feux pour un CAF.

Le nouveau plan peut influencer négativement les carrefours adjacents. Le contrôleur doit prendre en compte le voisinage entre carrefour, et si un conflit existe entre deux carrefours, le carrefour le plus saturé est privilégié (algorithme de la figure 5-14).

Calculer les PDF pour les carrefours concernés

S'il existe deux carrefours adjacents alors analyser leurs conflits

Privilégier le carrefour ayant le plus grande nombre de véhicules influencés

Figure 5-14 : Algorithme de calcul du nouveau plan de feux pour un CTRL et CDZ.

Le PC, jouant le rôle d'*Ordonnanceur*, fait passer les autres agents qui n'ont pas terminé leur exécution à la tranche horaire suivante, via le mécanisme de synchronisation. Si aucun problème n'a été détecté, la simulation se termine une fois la dernière tranche horaire traitée du dernier agent. Ce cas est représenté dans le diagramme de séquence AUML (figure 5-15). Par contre, lorsqu'un agent détecte un problème qu'il ne sait pas résoudre, il sollicite une aide extérieure (coopération). Le PC (informé ou sollicité) abandonne définitivement la partie ordonnancement pour se consacrer à la résolution du problème, puis termine la simulation.

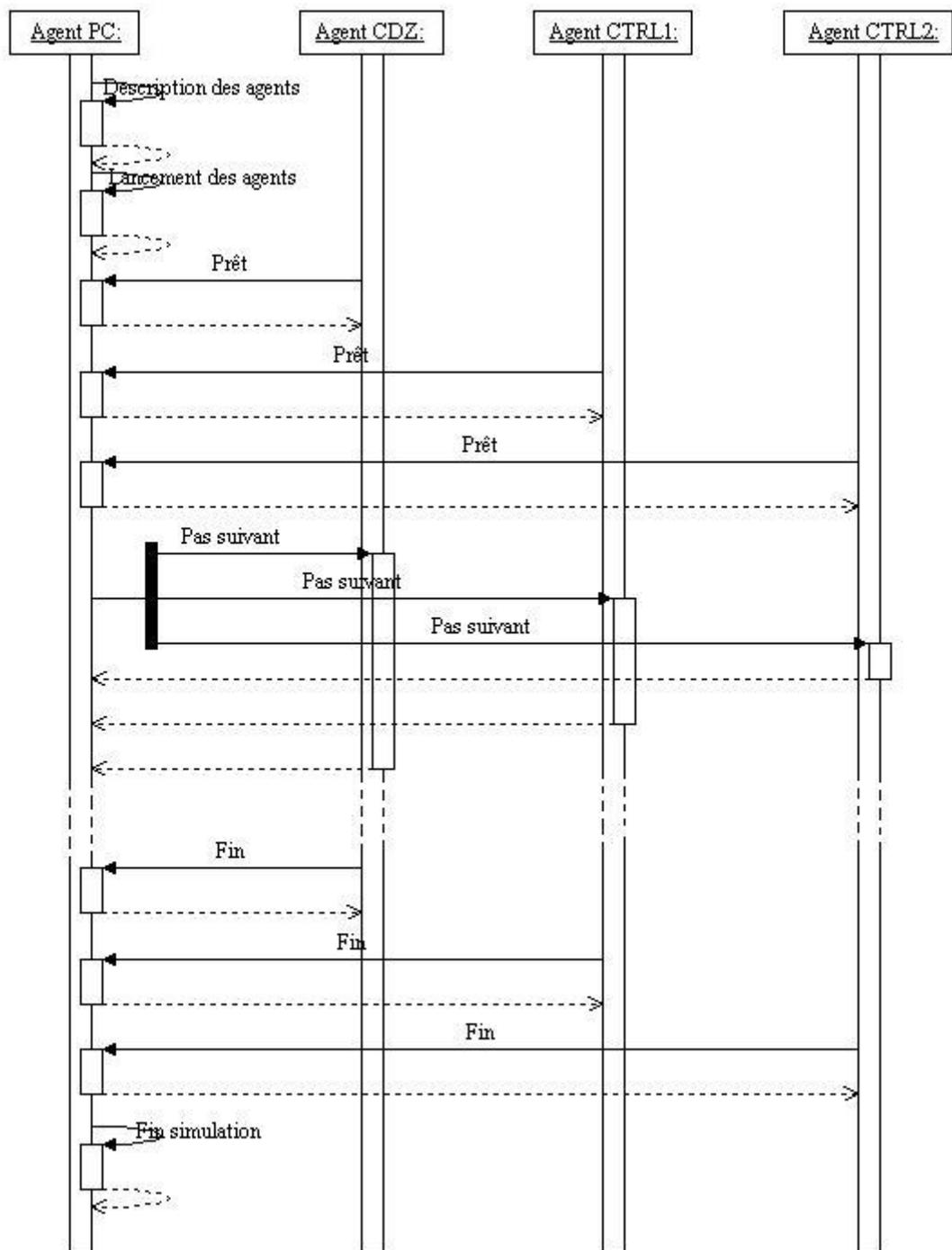


Figure 5-15 : Diagramme de séquence de synchronisation des agents ER

Les différents modes d'utilisation :

Notre SMA peut s'exécuter en trois modes :

1. Mode 1 : Couplage avec le modèle ASCImi-STU :

Dans ce cas, le SMA commence par l'exécution de l'agent IU, permettant à l'utilisateur d'accéder aux paramètres de configuration ou de lancer le module

Design_SSD pour vérifier la structure hiérarchique du système étudié. Ensuite, la simulation commence en lançant l'agent PC.

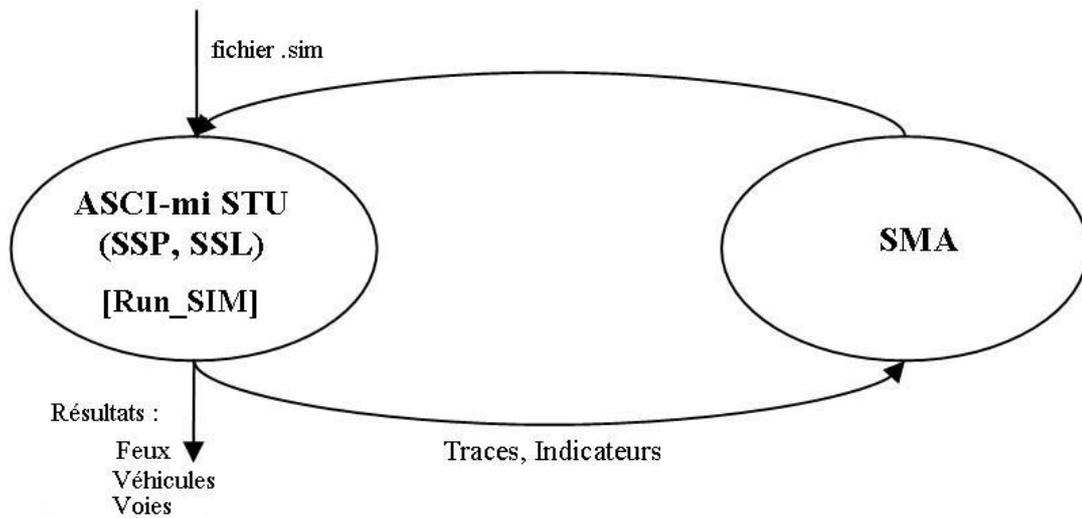


Figure 5-16 : Mode 1 - Couplage avec le modèle ASCImi-STU.

Dans ce mode, le SMA donne en sortie une liste des événements rencontrés ordonnés sur le temps des occurrences ou par ordre d'importance (par rapport aux entités influencées). Il donne aussi des propositions pour modifier les paramètres initiaux du module Run_SIM, comme la modification du phasage pour toute la période de simulation.

2. Mode 2 : Interactif avec le modèle ASCImi-STU :

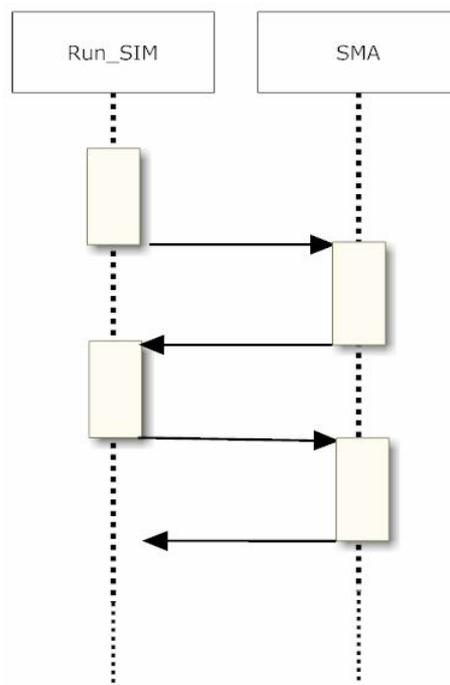


Figure 5-17 : Mode 2 : Interactif avec le modèle ASCImi-STU.

Le SMA intervient sur le déroulement de Run_SIM en modifiant ses paramètres au cours de la simulation. L'utilisation de plans de feux adaptatifs est simulée dans ce cas. Les contrôleurs prennent des décisions locales selon leurs privilèges. Les coordinateurs et le poste central ont des privilèges supérieurs et ils peuvent modifier le phasage des feux avec des marges plus grandes.

Le SMA dispose d'une option permettant au décideur d'intervenir à chaque étape dans le processus de simulation.

- a. Avec interprétation de l'utilisateur
- b. Sans interprétation de l'utilisateur

Communication entre agents :

Tous les agents jouent le rôle d'Entité de Régulation (ER) et d'autres rôles qui leur sont propres. Le rôle ER permet au poste central de synchroniser (via une gestion dynamique de la liste des agents) les autres ER, dans la mesure où elles ont chacune un comportement asynchrone.

Sur le plan de la communication entre agents, nous utilisons deux types de messages proposés dans la librairie MadKit :

- La classe 'StringMessage' est utilisée pour mettre en œuvre le mécanisme de synchronisation entre les agents. Pour les ordonnancer, le PC envoie un 'broadcast' message, et personne n'est ainsi privilégié. En retour, lorsqu'un agent a terminé de dérouler ses rôles, il informe le PC qu'il est prêt à poursuivre la simulation.
- Conformément aux préconisations de la FIPA, au titre de l'acte de langage, nous utilisons la classe 'ACLMessage'. Un tel message permet aux agents d'échanger des données sur le STU afin d'interagir entre eux (coopérer, ...).

Si un agent veut envoyer un message à un autre, il le cherche dans son groupe local, une fois trouvé, il lui passe le message directement. Si le récepteur n'appartient pas à son groupe il contacte son supérieur (le coordinateur de zone ou le poste central).

4-Modèle de résultats

Le modèle multi-agents proposé traite les problèmes du STU du point de vue des services techniques. Les critères de performance concernent la vitesse moyenne de la circulation, le temps d'attente aux carrefours, la taille et la durée des saturations, etc. Les résultats obtenus par le SMA dépendent du mode d'exécution choisi :

Le mode couplage : il permet d'enrichir la bibliothèque de plans de feux. Mais comme il change les paramètres de la simulation une fois, la performance du système s'améliore de façon à diminuer le nombre de saturation sur les sections critiques.

Le mode interactif : on simule la gestion du trafic en temps réel avec l'utilisation des plans de feux adaptatifs.

5- Conclusion

Nous avons présenté dans ce chapitre la réalisation d'un modèle décisionnel basé sur l'approche multi-agents.

La structure des carrefours (les lignes de feux, les courants et les voies) est obtenue grâce à la description physique du système. L'utilisateur doit définir les entités de régulation, (les contrôleurs et les coordinateurs de zone), et faire appartenir les CAF à leurs contrôleurs, les ER seront instanciées comme agents avec le noyau de MadKit.

Suite à l'adoption de l'approche multi-agents, l'environnement de modélisation ASCImi-STU a été enrichi. Nous avons introduit de nouveaux outils d'analyse et de spécification (AUML, MaSE, AgentTool). Nous avons construit et implémenté notre modèle en utilisant le langage Java sous la plateforme MadKit. La figure 5-13 montre l'environnement de modélisation ASCImi-STU étendu par l'approche multi-agents :

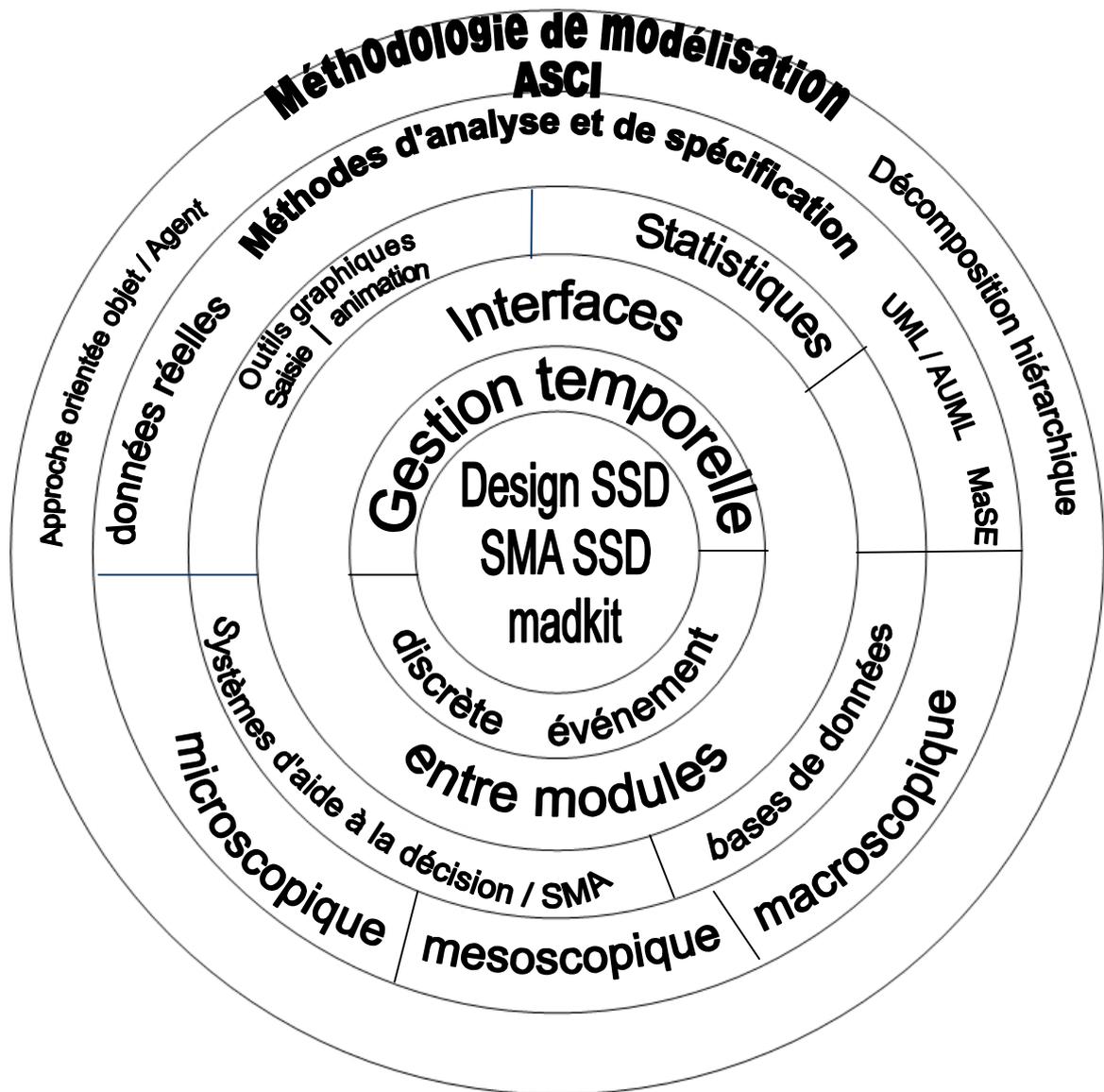


Figure 5-18 : L'environnement de modélisation ASCImi-STU SMA.

Chapitre 6

- Mise en œuvre

Sommaire

1- INTRODUCTION	131
2- CAS D'ÉTUDE : RIOM	132
<i>2-1- Objectifs et problèmes</i>	<i>133</i>
<i>2-2- Modèle de connaissance</i>	<i>133</i>
<i>2-3- Modèle d'action :</i>	<i>135</i>
<i>2-4- Modèle de résultats :</i>	<i>136</i>
3- CONCLUSION	147

1- Introduction

Ce chapitre est dédié à la mise en œuvre du modèle multi-agents du SSD sur un exemple réel « Ville de Riom ». La figure 6-1 présente un schéma général afin de réaliser un modèle d'action du système étudié.

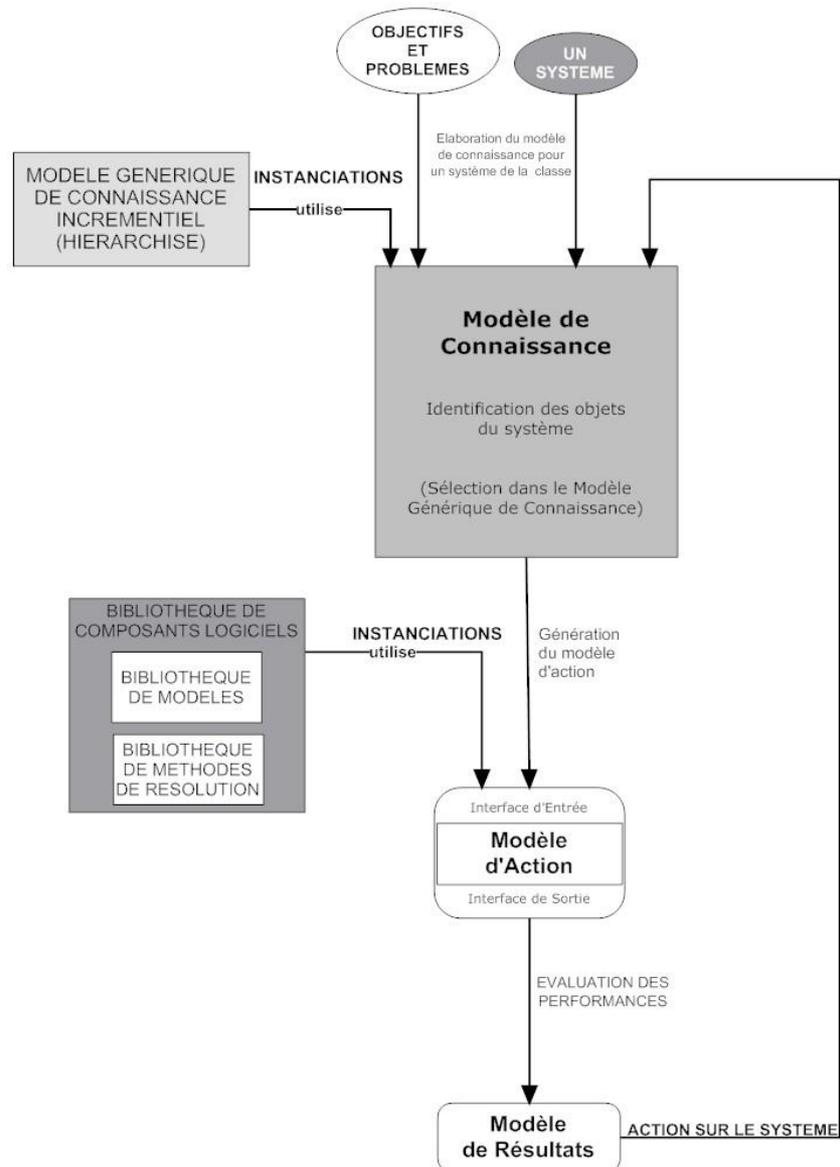


Figure 6-1 : Schéma de mise en œuvre.

Suite à la définition des objectifs et des problèmes à résoudre, le modèle générique de connaissance du domaine qui est fait dans la première phase de la méthodologie ASCImi-STU nous sert à construire le modèle de connaissance du système étudié. La bibliothèque de composantes logicielles obtenue en phase de conception sera instanciée afin de générer le modèle d'action.

2- Cas d'étude : Riom

Nous présentons un exemple de la ville de Riom afin de mettre en valeur les fonctionnalités du modèle SMA construit. La figure 6-2 montre un schéma de la ville Riom :

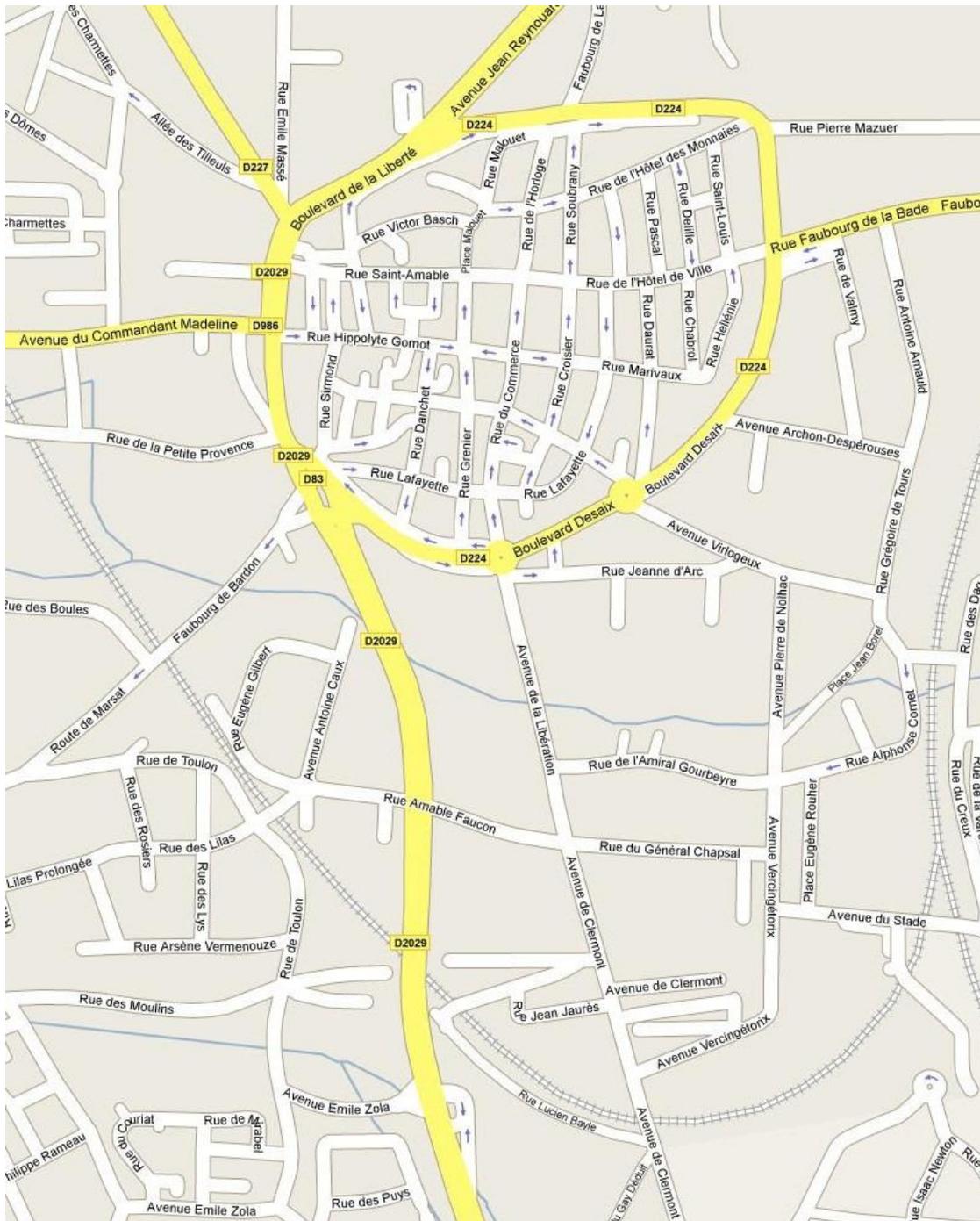


Figure 6-2 : Schéma de la ville Riom.

L'objectif est de décourager les automobilistes d'entrer dans Riom en favorisant la circulation sur la couronne, donc minimiser le temps de trajet nord-sud ou sud-nord, en optimisant les plans de feux. Nous détaillons les objectifs de l'étude et les zones prises en compte dans le modèle de connaissance.

2-1- Objectifs et problèmes

L'objectif de ce modèle de pilotage est dans un premier temps d'analyser et d'évaluer la performance du système. Les critères retenus sont la vitesse moyenne des véhicules, le temps d'attente sur les carrefours, la longueur et la durée des saturations. Le deuxième objectif est de prendre des décisions afin d'améliorer le fonctionnement du système par rapport à ces critères.

Puisque on parle du pilotage, nous sommes dans le niveau opérationnel par rapport à l'horizon temporel. Les actions prises à ce niveau concernent essentiellement le changement du phasage des feux à plusieurs niveaux (niveau local : les contrôleurs, niveau intermédiaire : les coordinateurs de zones et niveau global : le poste central).

2-2- Modèle de connaissance

La figure 6-2 montre le modèle physique du réseau de la ville Riom, la description de ce modèle existe dans un fichier *graphe.xml*. Le passage *quasi-automatique* de la description physique vers la description décisionnelle dont nous avons parlé dans le chapitre précédent nous amène au fichier *ssd.xml*, où on trouve dix carrefours à feux contenant 22 lignes de feux et 77 voies. L'étape suivante est l'utilisation de l'outil *design_SSD* afin de vérifier la structure des courants et de définir les entités de régulation type : poste central, coordinateurs de zone et contrôleurs, et ensuite d'attribuer les carrefours à feux aux contrôleurs. (Ces entités de régulation ne correspondent pas à la réalité, ce sont des entités virtuelles. Les matériels utilisés dans la ville de Riom ne supportent pas la structure hiérarchique proposée et n'utilisent pas de plans de feux adaptatifs). Nous définissons deux coordinateurs de zone comme suite (figure 6-3):

Zone sud : un coordinateur est associé à cette zone, il gère deux contrôleurs :

CTRL1 responsable des carrefours N100 et N101

CTRL2 responsable des carrefours N7 et N26

Zone nord : le coordinateur associé à cette zone gère trois contrôleurs :

CTRL3 responsable du carrefour N104

CTRL4 responsable des carrefours N105, N106 et N107

CTRL5 responsable du carrefour N108

Un contrôleur isolé gère le carrefour N47

L'ensemble des contrôleurs et les deux coordinateurs de zone sont liés au poste central.

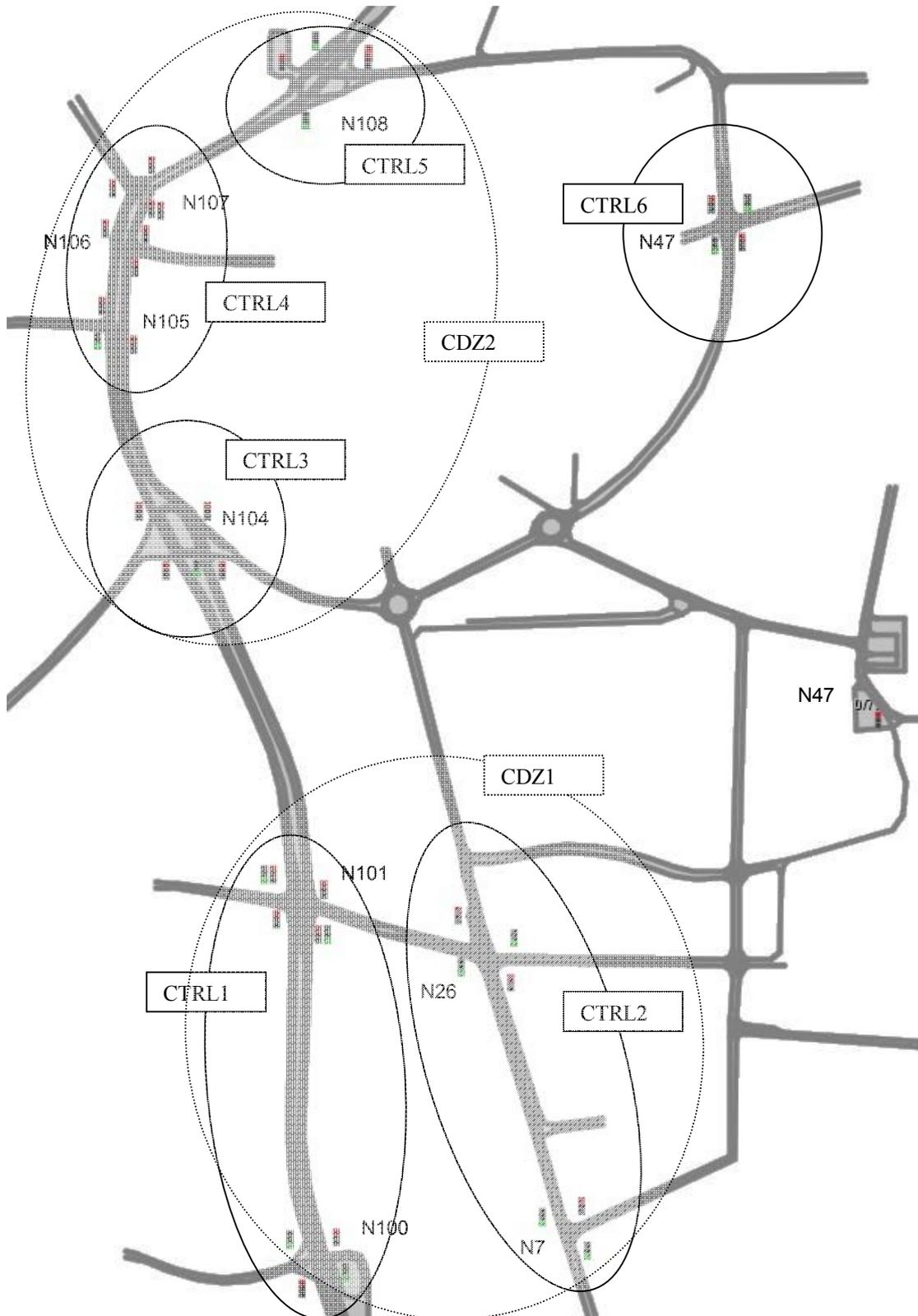


Figure 6-3 : Schéma du sous-système physique de la ville Riom avec les configurations décisionnelles associées.

2-3- Modèle d'action :

La bibliothèque des composants logiciels obtenue dans la phase de conception contient des classes qui représentent toutes les entités du domaine. Dans ce modèle, on utilise essentiellement les classes d'agents : l'interface utilisateur (IU), la base de données (BD) et les agents type entités de régulations : le poste central (PC), les coordinateurs de zones (CDZ) et contrôleurs (CTRL). D'autres composants comme les carrefours à feux, lignes de feux, courants sont utilisés. On y trouve également des entités physiques (voies, capteurs) et logique (feux tri-couleurs).

Le SMA_SSD est initialisé en instanciant l'agent IU qui lance à son tour l'agent BD et crée des objets à partir des classes du SSD en fonction de la description décisionnelle qui existe dans le fichier *ssd.xml*.

Les agents, en fonction de leur rôle, (voir section 4-4 dans le quatrième chapitre) cherchent à réaliser leurs objectifs. On peut identifier quatre grands modules : **surveillance, analyse, décision et communication**:

L'agent BD qui joue le rôle de **surveillance** prend en entrée les fichiers *Véhicule* et *Voie* afin de distribuer aux contrôleurs les fichiers de comptages qui les concernent.

- **La surveillance** du système étudié amène à deux fichiers de comptage pour chaque courant :
 - 1) Comptage temps réel : pour chaque instant, donne le nombre de véhicules existant sur le courant.
 - 2) Comptage cumulatif : donne le nombre de véhicules passant par le courant depuis le début de la période étudiée.
- Suite à l'**analyse** de l'ensemble des fichiers de comptages, les contrôleurs obtiennent une liste d'événements, un événement est défini par un triplet : le lieu d'événement, l'instant t auquel l'événement commence, et la durée de l'événement :

[Lieu, t, durée]

Les lieux peuvent être classés dans une liste de priorités définie selon le nombre et la durée totale des événements rencontrés, un élément de cette liste est défini par un triplet :

[Lieu, nombre d'événements, durée totale]

- **La prise de décision** se fait en trois niveaux : local (les contrôleurs), intermédiaire (les coordinateurs) et global (le poste central), prenant en compte les priorités des carrefours selon la liste de priorité obtenue précédemment, avec les informations des fichiers de

comptages et les contraintes de la maîtrise de sécurité de chaque carrefour. Les actions prises concernent essentiellement les plans de feux, un ensemble de nouveaux plans de feux est obtenu dans cette étape. Ces plans seront utilisés soit pour recommencer une deuxième simulation (mode 1) soit pour modifier les paramètres de la simulation en cours (simuler le pilotage en temps réel) (mode 2).

- **La communication** est réalisée parallèlement avec les trois modules précédents afin d'échanger les résultats de chaque étapes entre les agents concernés. L'agent BD envoie les fichiers de comptage aux contrôleurs. Les contrôleurs envoient la liste d'événements, la liste des priorités, une proposition de plans de feux et des fichiers de comptages agrégés (comptage par carrefour) aux coordinateurs. Les coordinateurs calculent et appliquent les plans de feux en fonction de leurs priorités. Les coordinateurs de zone réalisent des comptages par section pour l'envoyer au poste central.

2-4- Modèle de résultats :

Chaque module fournit des résultats à un autre module, parmi ces résultats on trouve :

- **Comptage** : un ensemble de fichiers de comptage à trois niveaux
- **Evénements** : un ensemble de listes d'événements, de listes de priorités
- **Actions** : un ensemble de plans de feux à appliquer aux carrefours
- **Proposition** : un ensemble de propositions de plans de feux à l'entité supérieure ou à l'utilisateur
- **Statistiques** : une comparaison entre le système initial et le système obtenu ou piloté. Cette comparaison comprend le nombre de véhicules servis, la vitesse moyenne des véhicules, et la somme du temps d'attente des véhicules.

Les résultats du SMA dépendent du mode de fonctionnement choisi (voir chapitre 5 précédent). Nous détaillons un exemple en mode couplage (mode 1), et un exemple en mode interactif (mode 2) :

2-4-1- Mode 1 : Couplage

Ce mode représente une structure centralisée où la prise de décision se passe au niveau du poste central. Nous prenons deux exemples, chacun est pour une période de deux heures, de 7h30 jusqu'à 9h30, et de 17h00 jusqu'à 19h00. Nous détaillons le processus de raisonnement pour chaque exemple.

2-4-1-1- Période : 7h30 – 9h30:

Un cas de saturation est défini par une durée critique à ne pas dépasser et un seuil de saturation. On définit un cas de saturation avec une durée égale à la moitié de la durée du cycle des plans de feux sur chaque carrefour, et un seuil de 70% (égale à 14 véhicules sur une voie capturée d'une longueur de 100m). Suite à la surveillance et à l'analyse des fichiers de comptage, le SMA détecte un ensemble d'événements.

Chaque contrôleur regroupe ses événements selon le lieu de leurs occurrences. Par conséquent, il identifie les carrefours les plus importants (la liste de priorité) par rapport au nombre d'événements et à leurs durées totales :

Evénements détectés du CTRL5 :

Carrefour	LDF	Courant	Instante de fin	durée
N108	Plan1	A131	2643	50
N108	Plan1	A139	4088	47
N108	Plan1	A139	6789	51

Tableau 6-1 : Evénements détectés par CTRL5

Tableau de priorité chez CTRL5 :

Carrefour	LDF	Courant	Nb. de cas	Σ durée
N108	Plan1	A131	1	50
N108	Plan1	A139	2	98

Tableau 6-2 : liste de priorité pour CTRL5

De la même manière, les autres agents CTRL envoient leurs listes d'événements à leur supérieur (l'agent CDZ). A leur tour, les CDZ envoient ces listes à l'agent PC (le poste central). L'agent PC reçoit une liste de tous les événements du système. Une représentation de ces événements se trouve sur la figure 6-4 :

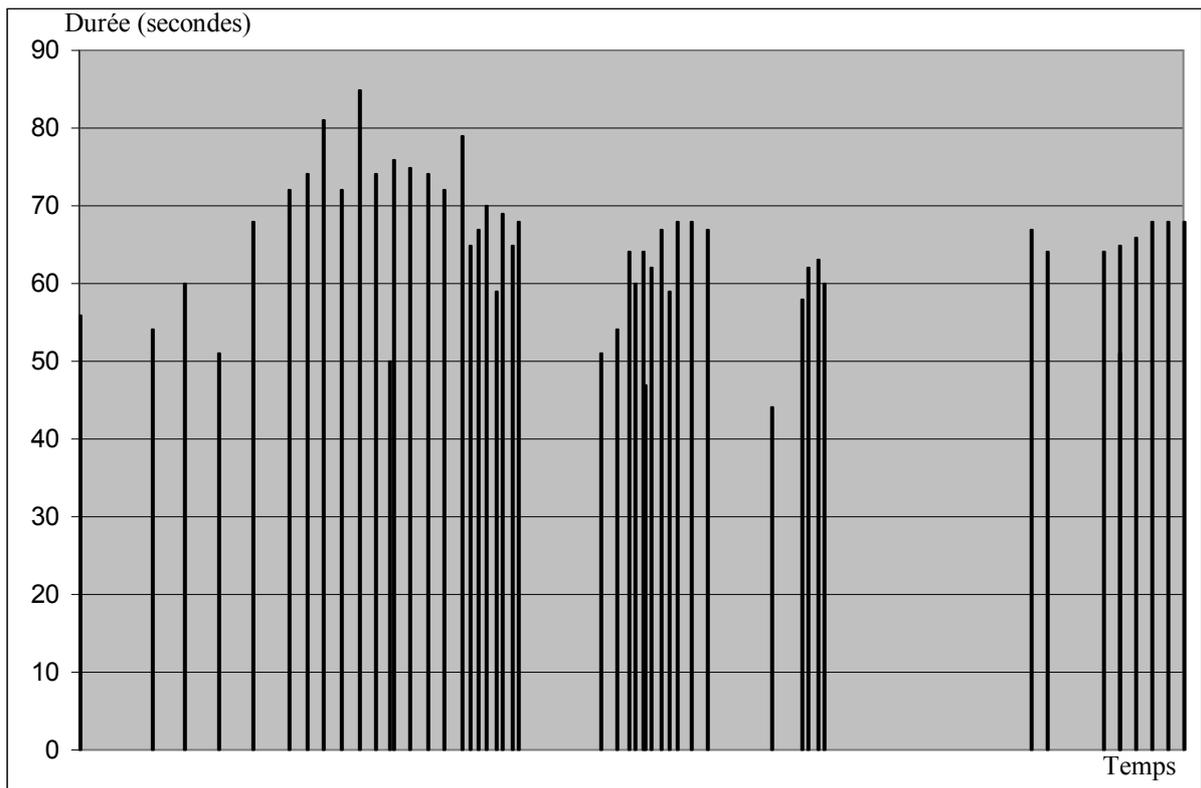


Figure 6-4 : Nombre et durée des événements pour la période d'étude.

L'agent PC établit la liste de priorité suivante :

CDZ	CTRL	Carrefour	LDF	Courant	Nb. de cas	Σ durée
Sud	CTRL1	N101	RN9	A112	26	1739
Nord	CTRL4	N107	Plan1	A130	20	1304
Nord	CTRL5	N108	Plan1	A139	2	98
Nord	CTRL5	N108	Plan1	A131	1	50
Nord	CTRL4	N105	RN9	A116	1	44

Tableau 6-3 : Liste de priorité selon les événements entre 7h30 et 9h30 pour une durée critique = cycle/2 et seuil = 70%.

Les décisions prises par le PC se basent sur cette liste de priorité. La modification de la durée de vert et de rouge pour les plans de feux concernés doit correspondre à la durée totale d'attente sur les carrefours. La durée de cycle des plans de feux sur les carrefours de Riom varient entre 58 secondes et 98 secondes. La durée effective totale de vert pour un carrefour est inférieure à la durée du cycle (en considérant les durées de jaune et les matrices de sécurité de chaque carrefour, voir section 4-3 Chapitre 4). Par exemple, le carrefour N101 (figure 6-5) a une durée de cycle égale à 98 secondes dont 80 secondes du vert distribuées entre trois lignes de feux :

- 1) La ligne *RN9* est composée des courant A107 et A112 (directions tout droit (TD) et tourner à droite (TAD)), elle a 30 secondes de vert.
- 2) La ligne *Faucon* est composée des courant A111 et A108, elle a 30 secondes de vert.
- 3) La ligne *RN9G* est composée des courant A107 et A112 (direction tourner à gauche (TAG)), elle a 20 secondes de vert.

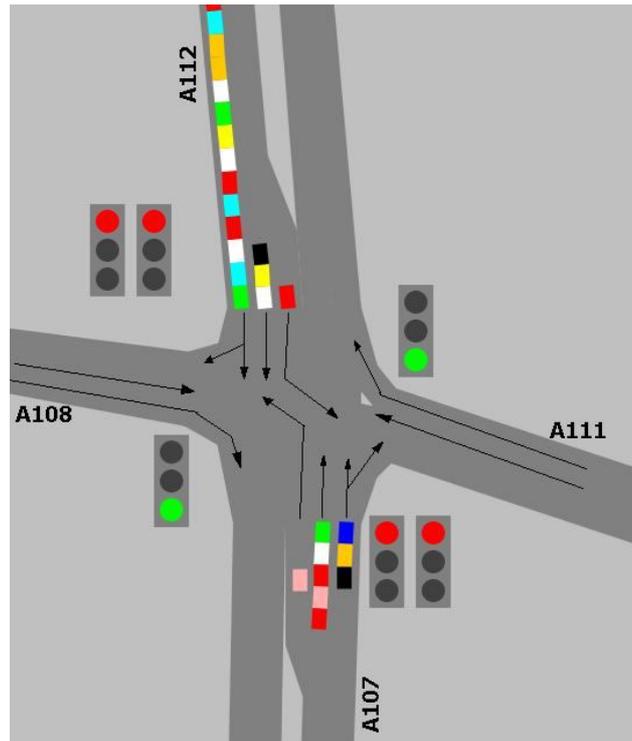


Figure 6-5 : La structure du carrefour N101

Selon la liste des priorités obtenue par l'agent PC, le carrefour N101 a une priorité supérieure, et l'action requise correspond à l'augmentation de la durée du vert de la ligne RN9. La valeur de l'augmentation dépend de la durée totale de vert (80 secondes) et de la matrice de sécurité de ce carrefour. Selon cette matrice les lignes *RN9G* et *Faucon* doivent avoir une durée de vert supérieur ou égale à 15 secondes, ce que nous amène à une durée maximale de verte pour la ligne *RN9* égale à 50 secondes. La figure 6-6 montre une représentation graphique des plans de feux du carrefour N101 et le tableau 6-4 montre les configurations originales et les nouvelles configurations de ces plans :

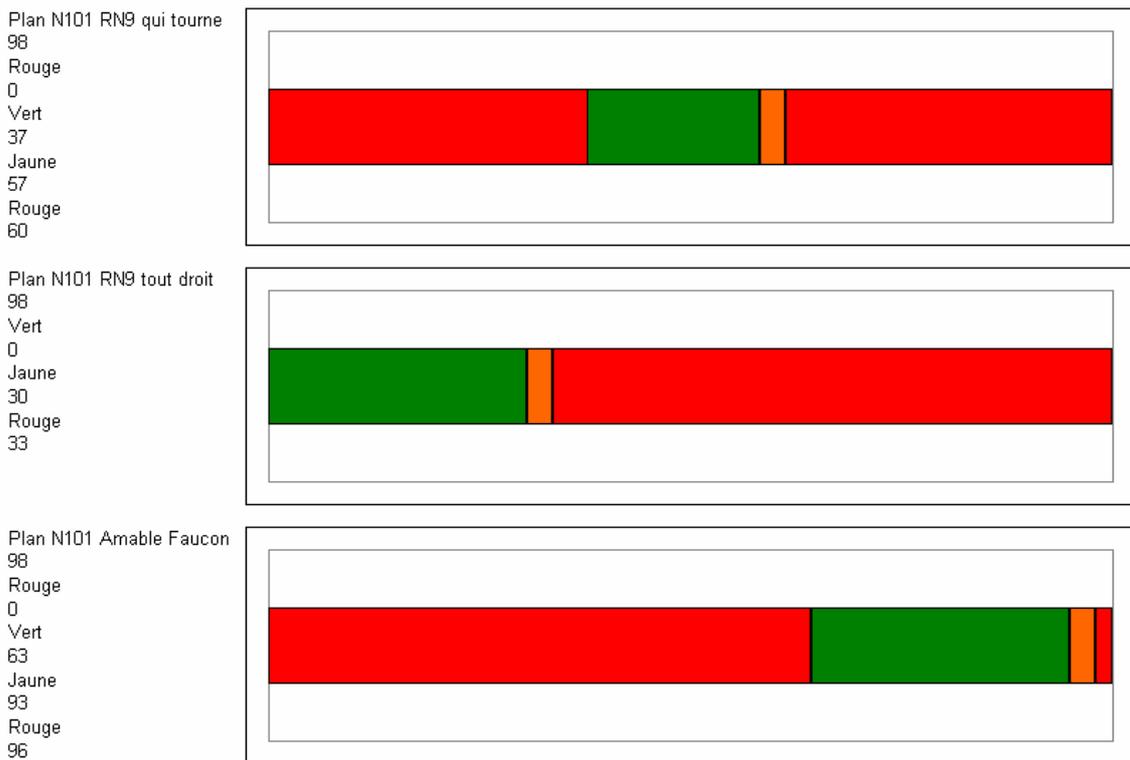


Figure 6-6 : Les plans de feux originaux du carrefour N101.

(Des représentations graphiques des plans de feux sur les autres carrefours se retrouvent dans l'annexe D).

Configurations originales			Nouvelles configurations		
<u>RN9</u>	<u>RN9G</u>	<u>Faucon</u>	<u>RN9</u>	<u>RN9G</u>	<u>Faucon</u>
3	4	4	3	4	4
98	98	98	98	98	98
Vert	Rouge	Rouge	Vert	Rouge	Rouge
0	0	0	0	0	0
Jaune	Vert	Vert	Jaune	Vert	Vert
30	37	63	50	57	78
Rouge	Jaune	Jaune	Rouge	Jaune	Jaune
33	57	93	53	72	93
	Rouge	Rouge		Rouge	Rouge
	60	96		75	96

Tableau 6-4 : Les plans de feux originaux et les plans de feux calculés pour le carrefour N101.

Les autres éléments de la liste de priorité (Tableau 6-3) sont traités de la même manière et relativement à la première modification (celle concernant le carrefour N101). Une augmentation de 15 secondes sera effectuée sur le vert de la ligne *plan1* du carrefour N107 et 1 seconde supplémentaire sur le vert des lignes *plan1* du carrefour N108 et *RN9* du carrefour

N105. Une modification d'une seule seconde n'amène pas à une amélioration intéressante et donc on peut ignorer ces modifications.

Le fait d'appliquer les nouvelles configurations sur le système étudié nous amène à un système meilleur dont la liste d'événements montre une diminution important du nombre et de la durée totale des saturations détectées. La figure 6-7 montre les nouveaux résultats :

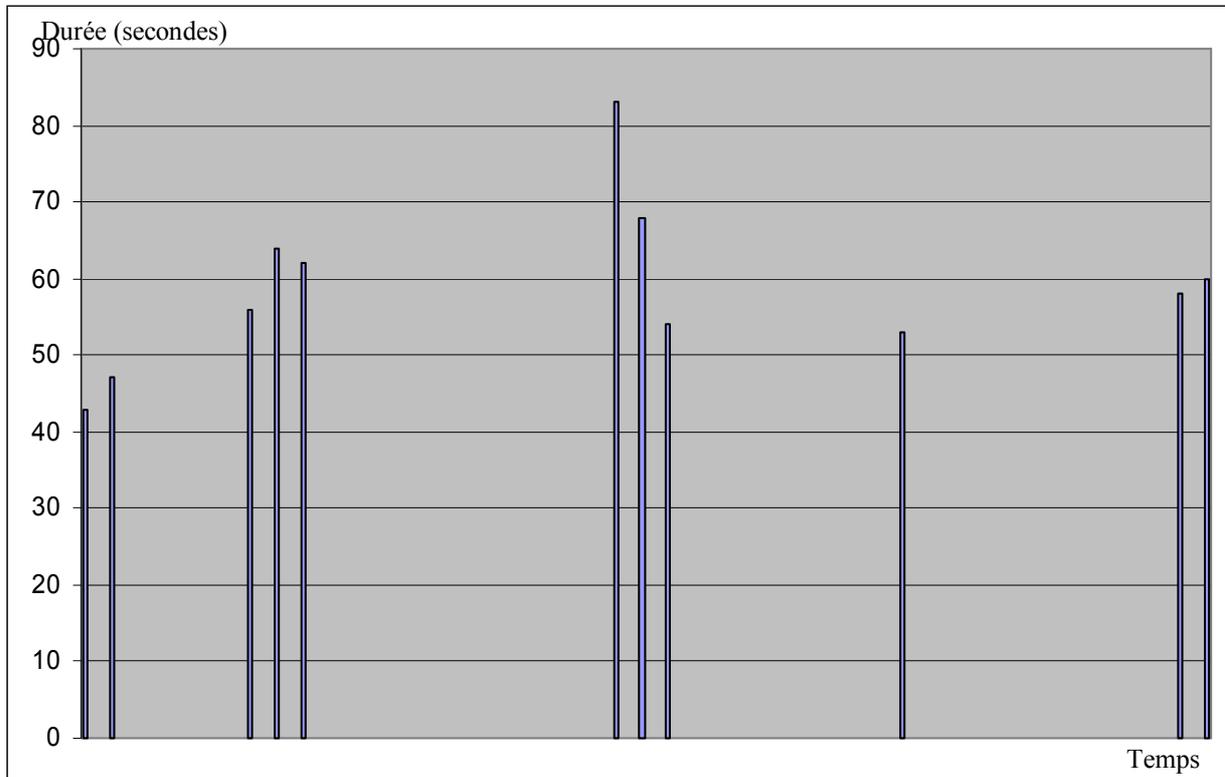


Figure 6-7 : Nombre et durée des événements pour la période d'étude suite à l'application des nouvelles configurations.

Nous comparons le système initial avec le système obtenu en considérant les critères : la vitesse moyenne des véhicules, le temps d'attente sur les carrefours, et le nombre de véhicules servis. Nous faisons cette comparaison sur les carrefours ayant eu des modifications importantes et aussi sur l'ensemble du système étudié. Le tableau 6-5 montre une comparaison entre les deux cas sur les carrefours N101 et N107. On peut remarquer une amélioration de la vitesse moyenne des véhicules passés par ces deux carrefours (de 26 vers 34 km/h) et une diminution du temps d'attente totale.

	N101		N107	
	Avant	Après	Avant	Après
Nombre de véhicules	3215	3213	2617	2627
Vitesse Moyenne (Km/h)	27	34.21	26	34
Temps total d'attente (secondes)	115635	63780	78853	28862

Tableau 6-5 : Comparaison entre le système initial et le système obtenu sur les carrefours modifiés.

La performance du système devient plus cohérente, des cas extrêmes sont éliminés et le temps d'attente diminue (Tableau 6-6) :

	Système initial	Système obtenu
Nombre de véhicules	9021	9021
Vitesse Moyenne (Km/h)	57.42	58.52
Temps total d'attente	485494	392557

Tableau 6-6 : Une comparaison globale entre le système initial et le système obtenu.

2-4-1-2- Période : 17h00 – 19h00:

Dans cet exemple nous prenons les mêmes critères que pour l'exemple précédent mais pour une période différente de 17h00 jusqu'à 19h00. Dans cet exemple nous démontrons l'utilisation des fichiers de comptages par les agents afin de calculer un nouveau plan de feux. La figure 6-8 montre les événements rencontrés dans cette période et le tableau 6-8 résume la liste de priorités obtenue.

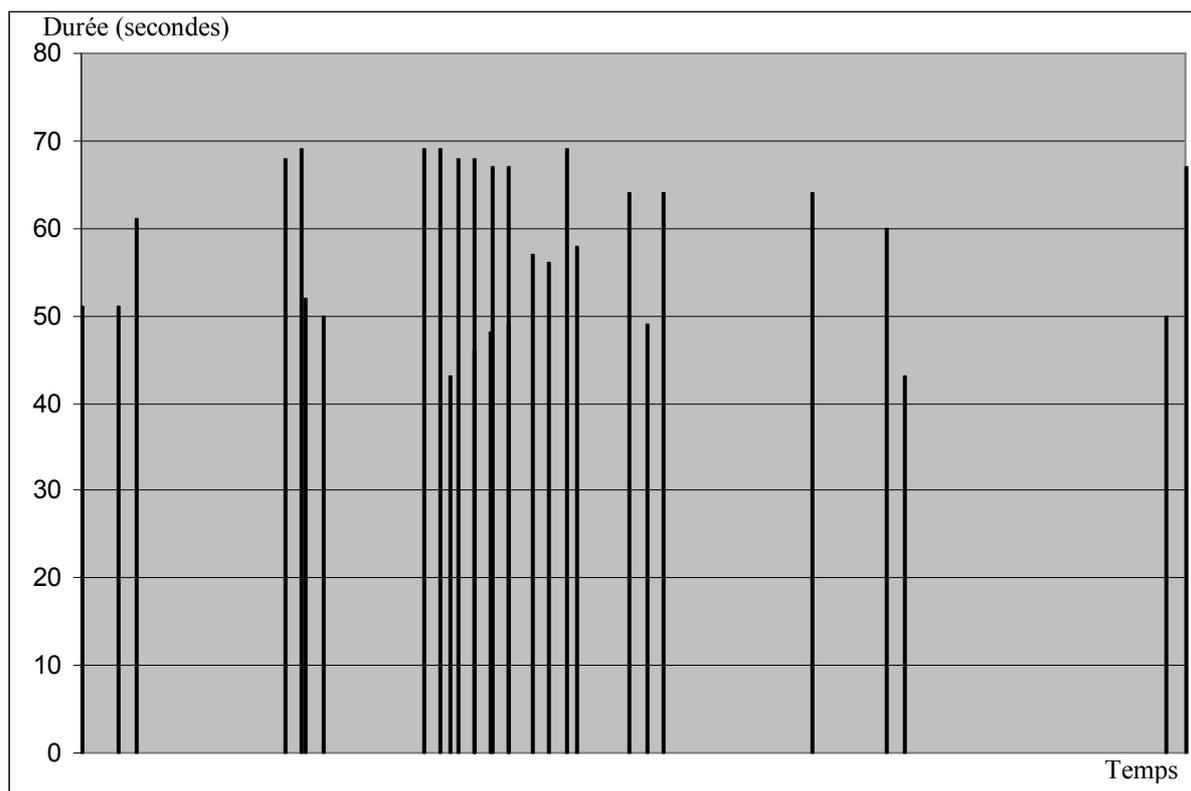


Figure 6-8 : Nombre et durée des événements entre 17h00 et 19h00.

Zone	Contrôleur	Carrefour	LDF	Courant	Nb. de cas	Σ durée
Sud	CTRL1	N101	RN9	A112	11	628
Nord	CTRL4	N107	Plan1	A130	9	585
Nord	CTRL4	N106	Plan1	A123	5	292
Sud	CTRL1	N101	RN9	A107	1	60
Nord	CTRL5	N108	Plan1	A131	1	46
Nord	CTRL4	N105	RN9	A116	1	43
Nord	CTRL4	N105	RN9	A122	1	43

Tableau 6-7 : Liste de priorités obtenues

Les comptages des courants pour cette période figurent dans le tableau 6-8, pour chaque carrefour, nous avons une liste de ses courants et le nombre total de véhicules servis.

N101		N105		N106		N107		N108	
A112	1577	A116	948	A123	897	A130	949	A131	1090
A112G	216	A122	955	A127	994	A107	851	A139	743
A107	1129	A120	814	A125	0	A128	783	A133	980
A107G	169							A141	153
A108	106								
A111	300								

Tableau 6-8 : Comptage cumulatif des courants des carrefours prioritaires

Pour chaque ligne de feux on identifie le courant ayant le nombre le plus élevé, nous obtenons par exemple pour le carrefour *N101* : (*A112:1577*, *A111:300*, *A112G:216*). Sachant qu'il y a une durée minimale de verte (selon la matrice de sécurité), une durée de 50 secondes de verte est associée à la ligne *RN9* (du courant *A112*), et 15 secondes pour les lignes *RN6G* et *Faucon*.

Le deuxième élément de la liste de priorité est celui de *plan1* du carrefour *N107*, mais en fonction du comptage est celui de *plan1* du carrefour *N106* donc une augmentation de 20 secondes est effectuée sur le vert de la ligne *plan1* du carrefour *N106* et 16 secondes est effectuée sur le vert de la ligne *plan1* du carrefour *N107*.

Suite à la mise en place de ces nouvelles configurations, les cas de saturation selon le critère prédéfini ont disparue. Des nouveaux événements ont été détectés pour un seuil de 40% et une durée égale à $\text{cycle}/2$. Une représentation de ces événements se trouve sur la figure 6-9.

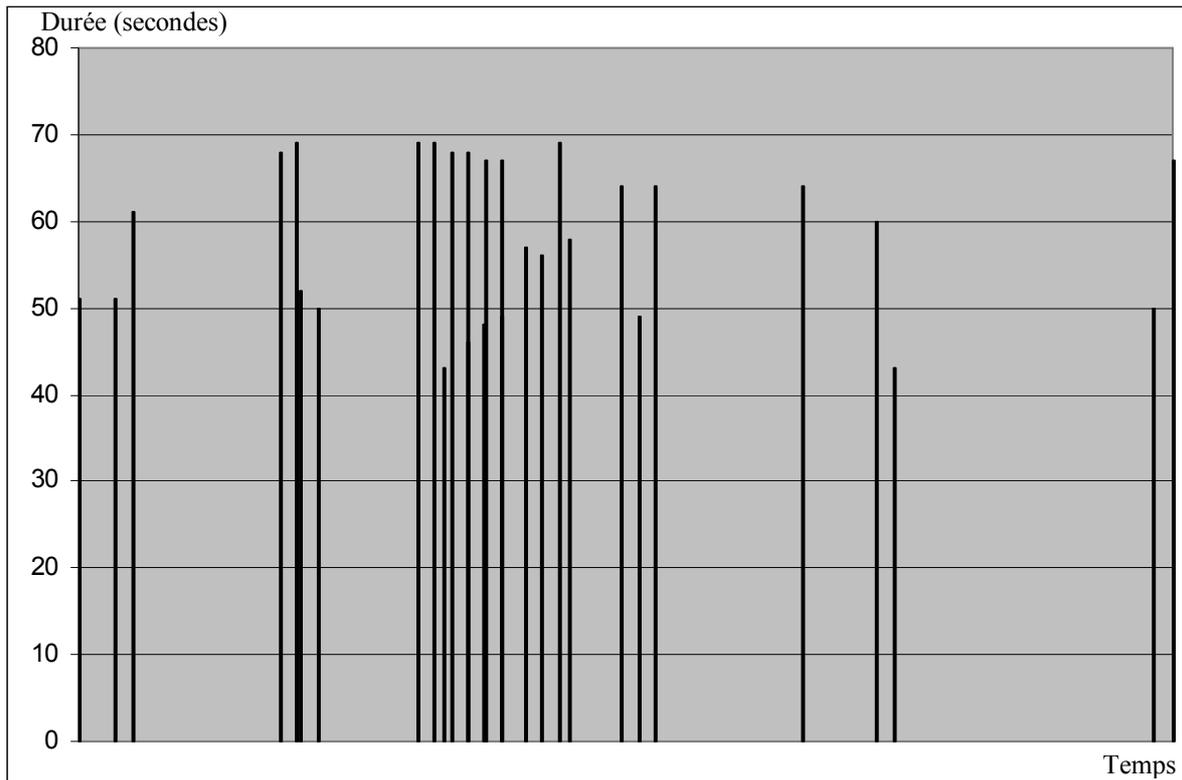


Figure 6-9 : nombre et durée des événements entre 17h00 et 19h00 pour un seuil = 40% et une durée = cycle/2.

Nous obtenons une liste d'événements beaucoup moins grave (tableau 6-9).

Zone	Contrôleur	Carrefour	LDF	Courant	Nb. de cas	Σ durée
Sud	CTRL2	N7	ClermontC	A9	1	39
Sud	CTRL2	N26	Liberation	A51	2	65
Sud	CTRL1	N100	RN9	A106	5	172
Sud	CTRL1	N101	RN9	A112	4	216
Sud	CTRL1	N101	Faucon	A111	1	50
Nord	CTRL3	N104	Plan2	A117	1	55
Nord	CTRL3	N104	Plan1	A115	3	145
Nord	CTRL4	N105	RN9	A122	1	46
Nord	CTRL4	N107	Plan1	N107	2	100
Nord	CTRL5	N108	Plan2	A131	1	47

Tableau 6-9 : Nouvelle liste de priorité en fonction des événements pour un seuil = 40% et une durée = cycle/2.

Une comparaison entre le système initial et le système obtenu en fonction de performance générale du système se trouve dans le tableau 6-10 :

	Système initial	Décisions globales
Nombre de véhicules	10052	10053
Vitesse Moyenne (Km/h)	59.91	61.96
Temps total d'attente	504834	318358

Tableau 6-10 : Comparaison entre le système initial et le système obtenu.

La figure 6-10 montre une comparaison du comptage sur *A112* avant les nouvelles configurations (Série 1, en bleu foncé) et après (Série 2, en rose):

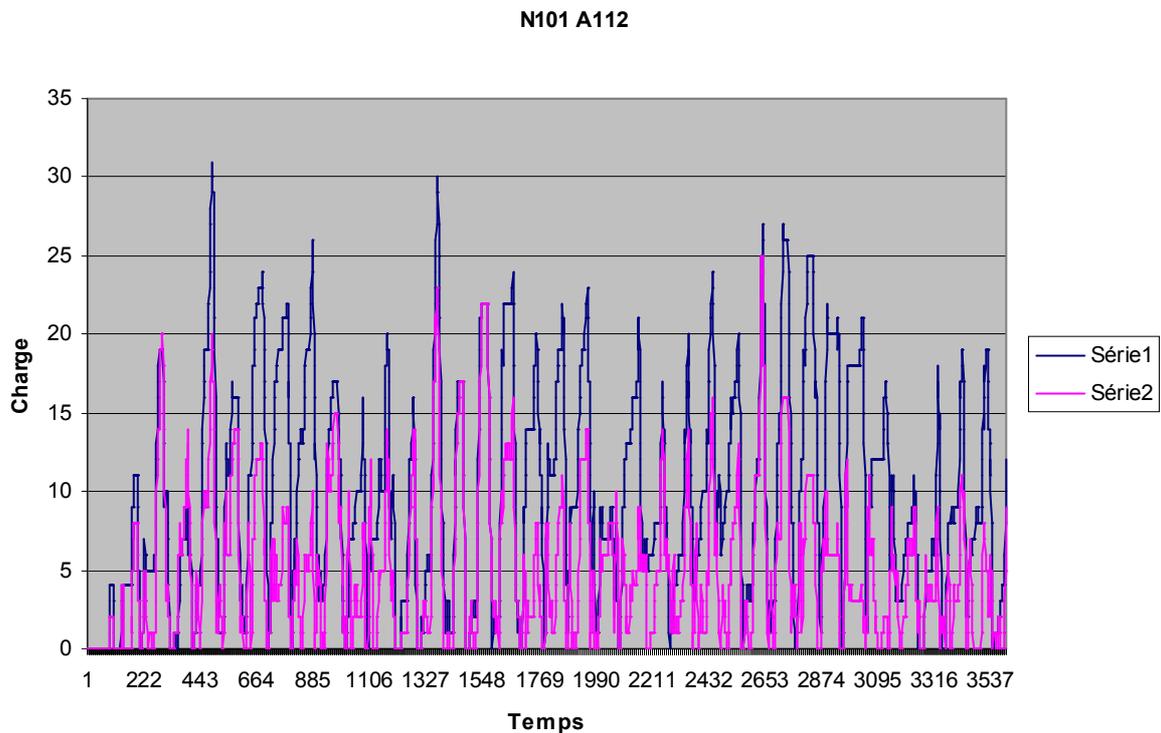


Figure 6-10 : Temps d'attente sur A112 avant : 50579, après : 26121, presque le même comptage cumulatif.

Le processus de raisonnement utilisé par les agents afin de trouver la bonne solution dans les deux exemples précédents est primitifs et prend en compte des statistiques du système pour la période d'étude et il propose une solution globale afin d'éliminer les cas extrêmes rencontrés. Ce cas correspond à l'enrichissement de la bibliothèque de plans de feux utilisée par les contrôleurs pour mieux adapter le système et configurer la table de commutation.

Dans l'exemple suivant nous détaillons le processus de raisonnement du pilotage en temps réel où le système doit réagir aux changements du système et proposer une solution chaque fois qu'un cas extrême est détecté. Ce mode représente l'utilisation des plans de feux dynamiques gérés en temps réel par les contrôleurs par rapport aux changements de la charge sur les courants.

2-4-2- Mode 2 : interactif

Dans ce cas on simule le pilotage en temps réel. Lorsque un problème est détecté à un instant t , le contrôleur concerné analyse les fichiers de comptages disponible à l'instant t afin d'évaluer la situation sur les carrefours adjacents relativement au problème détecté.

L'instant de mise en œuvre de la décision prise est l'instant de début du prochain cycle de phasage sur le carrefour concerné.

Exemple : période 17h00 – 18h00

On définit un cas de saturation avec une durée égale à la durée du cycle des plans de feux sur chaque carrefour, et un seuil de 70%.

Le premier événement se passe à l'instant $t = 2181$ sur le courant *A116*, carrefour *N105*. Le comptage des courants du carrefour : *A116*, *A122* et *A120* (*A116* et *A122* ont la même ligne de feux) montre un équilibre entre les trois courants, mais à l'instant t le courant *A116* a une charge trois fois supérieure à celle du courant *A120*.

	$t = 2181$	Totale [0-2181]
A116	70%	331
A122	0	338
A120	25%	367

Tableau 6-11 : comptage sur N105

L'agent *CTRL4* récupère les fichiers de comptage du carrefour *N106* adjacent afin d'évaluer la situation sur la *N106* relativement à celle de la *N105*, un cas similaire existe sur la *N106* mais moins grave. La décision prise par l'agent *CTRL4* est : 5 secondes supplémentaires du vert sur la ligne *RN9* du carrefour *N105* et 3 secondes supplémentaires du vert sur la ligne *plan1* du carrefour *N106*.

Bilan :

Dans l'état actuel de ce travail, comme nous l'avons montré précédemment, le SMA détecte les cas de saturations, évalue la performance du système et prend d'action aux trois niveaux afin d'améliorer la performance du système selon les critères prédéfinis. L'implémentation de ce mode d'utilisation est en cours, il manque de pouvoir appliquer les nouvelles configurations au cours de la simulation et continuer l'exécution de cette simulation en prenant en compte ces nouvelles configurations.

3- Conclusion

Dans ce chapitre, nous avons montré l'utilisation de notre modèle multi-agents de pilotage qui s'intègre dans le modèle de simulation ASCImi-STU

L'architecture réalisée permet d'évaluer la performance du système étudié selon des critères prédéfinis, et de prendre et de mettre en œuvre des décisions afin d'améliorer le fonctionnement du système. Elle fournit à l'utilisateur des données de différents types : comptages et statistiques, événements, proposition et actions autonomes. Elle permet entre autres de valider la répartition des plans de feux sur un carrefour.

L'approche multi-agents proposée permet de représenter, dans le modèle de simulation, l'autonomie des centres de décision des STU. Ces centres existent à plusieurs niveaux hiérarchiques de prise de décision. Ils doivent donc toujours faire un équilibre entre l'autonomie et la coopération avec le système externe.

Conclusion générale

Dans cette thèse, nous nous sommes intéressés au problème du pilotage des systèmes du trafic urbain. Nous avons proposé une démarche de modélisation en partant de l'analyse jusqu'à la conception et la mise en œuvre d'un modèle de pilotage des STU dans le but de concevoir une architecture permettant de tester différentes techniques de micro et macro régulation.

La dynamique d'un système complexe peut être représentée par des entités autonomes interagissantes. L'utilisation de l'approche multi-agents apparaît dans ce contexte comme un choix naturel. Cependant les démarches de conception restent liées au contexte du domaine.

Cette approche a été adoptée pour construire le modèle de pilotage. Le paradigme agent nous a paru approprié pour spécifier l'architecture et le comportement des trois équipements de régulation (contrôleur de carrefours à feux, coordinateur de zone et poste central). Nous avons associé un agent à chacune de ces trois entités décisionnelles dans la mesure où elle a pour objectif de maintenir le STU étudié dans un état fluide. Le SMA ainsi obtenu représente le système réel d'une manière aussi claire et réaliste que possible.

Les travaux présentés se sont déroulés dans le cadre du développement d'un environnement de modélisation et de simulation des STU. Le point de départ de ces travaux est la méthodologie ASCImi-STU développée dans l'équipe MAD (Modélisation et Aide à la Décision) que nous avons étendue et enrichie par l'utilisation de l'approche multi-agents. Nous avons appelé cette extension ASCImi-STU SMA.

Originalité du travail

L'originalité de ce travail réside dans le développement d'un modèle multi-agents pour le pilotage des STU dans un cadre méthodologique (ASCImi-STU SMA). Cette méthodologie a permis d'enrichir l'environnement de modélisation dédié aux STU et d'intégrer le modèle décisionnel avec les autres modèles appartenant au domaine (les modèles physique et logique), prenant en compte une modélisation à plusieurs niveaux microscopique, mesoscopique et macroscopique,

Les éléments que nous avons intégrés dans la démarche méthodologique ASCImi-STU mise en place constituent l'ensemble de nos contributions au regard des outils et des modèles existants.

L'utilisation d'outils de conception dédiés à l'approche multi-agents a permis de concevoir l'aspect dynamique du système et d'ajouter des notions mentales (problème, objectif, ...) dans le modèle construit.

Apports spécifiques

Les travaux menés dans cette thèse ont contribué à plusieurs apports dédiés au domaine de modélisation des systèmes complexes et à l'utilisation de l'approche multi-agents :

- Introduire de nouveaux outils de spécification et de conception (MaSE) dans l'environnement de modélisation ASCImi-STU
- Présenter les entités décisionnelles par des agents et spécifier le comportement de chaque agent sous forme d'objectifs, de rôles, de tâches.
- Simuler le pilotage en temps réel par un modèle SMA qui est capable de détecter des problèmes et de réagir et mettre en œuvre d'actions appropriées. Le modèle construit a été implémenté sur la plateforme Madkit.
- Proposer un outil d'aide à la décision en facilitant l'interactivité home-machine, et fournir à l'utilisateur des informations, des explications et des suggestions.

Le dernier chapitre montre une application de la démarche méthodologique proposée au domaine de pilotage des STU. Cette démarche peut être appliquée à d'autres systèmes complexes, prenant en compte les aspects spécifiques à chaque domaine.

D'après le cadre d'évaluation que nous avons défini dans le chapitre 2 (2-5), les travaux réalisés dans cette thèse couvrent les quatre applications d'une entité de régulation, et donc nous nous situons au degré 6/6 au niveau conceptuel. Bien entendu que l'exemple que nous avons présenté dans le chapitre 6 ne prend pas en compte l'application VTC et l'instanciation de PMV est virtuelle à cause de l'absence de ces deux applications dans la ville Riom. Nous arrivons à un degré de 4/6 au niveau d'implémentation.

Par rapport à l'horizon temporel et spatial, nous avons commencé cette étude au niveau opérationnel et microscopique et nous avons avancé jusqu'au niveau tactique mesoscopique, autrement dit : nous couvrons quatre cases sur neuf du tableau 2-4 (p. 45).

Perspectives

Nous envisageons de nombreuses perspectives de ces travaux autour des différents aspects abordés durant cette thèse :

- Etudier d'autres exemples qui prennent en compte les autres applications : véhicules de transport en commun (VTC) et les panneaux aux messages variables (PMV)
- Enrichir la base de connaissance des agents (tâches classifiées et triées)
- Améliorer l'interface utilisateur et l'intégration avec Run_Sim

L'architecture proposée permet de tester différentes techniques de régulation microscopiques et mesoscopiques afin de résoudre des problèmes de planification aux niveaux tactique et opérationnel. L'architecture réactive mise en œuvre pourrait être étendue afin de tester des régulations macroscopiques prenant en compte les problèmes de planification au niveau stratégique. Elle pourrait également être améliorée afin de profiter de l'historique des actions mises en œuvre et du résultat obtenu.

Références bibliographiques

Adler, J. L. and V. J. Blue (2002). "A cooperative multi-agent transportation management and route guidance system." *Transportation Research Part C* 10: 433–454.

Algers, E., Bernauer, S., Boero, S., Breheret, L., di Taranto, C., Dougherty, M., Fox, K. and Gabard, J. (1997) "Review of micro-simulation models". Smartest Project Deliverable D3, Leeds, 1997.

Ali, A., Chabrol, C., Fleig, O., Gourgand, M., (2005). L'approche multi-agents pour le pilotage des Systèmes de Trafic Urbain. Journées Francophones sur les Systèmes Multi-Agents JFSMA05. Calais.

Ali, A., Chabrol, M., Gourgand, M., (2007). Multi-agents approach for modelling traffic managing systems - an architecture for a decision-making aid tool. 21st EUROPEAN Conference on Modelling and Simulation ECMS 2007, Prague, Czech Republic.

Bauer, B., Müller, J. P., Odell, J. (2001). Agent UML: A Formalism for Specifying Multiagent Software Systems. *International Journal of Software Engineering and Knowledge Engineering*.

Boero, M., Cuenca, J., Kirschfink, H., Krogh, C. (1993). KITS: a general approach for knowledge-based traffic control models. *Technical Days on Advanced Transport Telematics*, Bruxelles.

Boero, M., Cuenca, J., Kirschfink, H., Traetteberg, H., Wild, D., (1994). The role of knowledge-based models in traffic management and their design. In *Towards an Intelligent Transport System. First World Congress on Applications of Transport Telematics and Intelligent Vehicle-Highways Systems*, Paris.

Bretherton, R. D., et al (1990). Recent enhancements to SCOOT—SCOOT version 2.4. 3rd *International Conference on Road Traffic Control/IEEE*, London.

Bromarius, F. (1992). "A multi-agent approach toward modeling urban traffic scenarios." *DFKI-RR* 92-47.

Burmeister, B., A. Haddadi, et al. (1997). "Application of multi-agent systems in traffic and transportation." *IEE Proc. Software Engineering* 144(1): 51–60.

Chabrol, M., Sarramia, D., (2001). Modelisation multiple des systemes de trafic urbain 3eme Conférence Francophone de MODélisation et SIMulation "Organisation et Conduite d'Activités dans l'Industrie et les Services" MOSIM'01, Troyes (France).

Chabrol, M., Sarramia, D., (2003). Modelisation hybride et simulation combinee des systemes de trafic urbain. 4eme Conférence Francophone de MODélisation et SIMulation "Organisation et Conduite d'Activités dans l'Industrie et les Services" MOSIM'03, Toulouse (France).

Chabrol, M., Sarramia, D., Tchernev, N., (2005). "Urban traffic systems modelling methodology." *International journal of production economics* (Elsevier).

Chang, E. C., K. K. Ho, et al. (1995). "Development of an integrated freeway-management system." Proceedings of the Annual Meeting of the ITS America I(ITS America): 589–596.

Combes, C. (1994). Un environnement de modélisation pour les systèmes hospitaliers. LIMOS. Clermont-Ferrand, Université Blaise Pascal.

Crawford, W., Kaplan, J., (2003). J2EE Design Patterns. O'Reilly.

Cuena, J., Martin, G., Molina, M. (a) (1992). An architecture for knowledge-based traffic management for the Expo-92. Sevilla Urban Ring, OECD.

Cuena, J., Ambrosino, G., Boero M., (b) (1992). A general knowledge-based architecture for traffic control: KITS approach. Artificial Intelligence Applications in Transportation Engineering, San Buena- Ventura, CA.

Cuena, J., J. Hernandez, et al. (1995). "Knowledge-based models for adaptive traffic management systems." Transportation Research 3(5): 311-337.

Cuena, J. and M. Molina (2000). "The role of knowledge modeling techniques in software development : a general approach based on a knowledge management tool." Human-Computer Studies 52: 385-421.

Davol , A. P. (2001). Modeling of Traffic Signal Control and Transit Signal Priority Strategies in a Microscopic Simulation Laboratory. Providence, RI, Brown University.

Demazeau, Y. (1995). From interactions to collective behaviour in agent-based systems. the First European conference on cognitive science ECCS '95.

Demazeau, Y. (2001). VOYELLES. Habilitation A Diriger des Recherches. Grenoble, Institut National Polytechnique de Grenoble INPG.

Drogoul, A., Collinot, A., (1998). "Applying an Agent-Oriented Methodology to the Design of Artificial Organizations : A Case Study in Robotic Soccer." Journal of Autonomous Agents and Multi-Agent Systems 1(1): 113-129.

Drogoul, A., Vanbergue, D., Meurisse, T., (2002). Multi-Agent Based Simulation : Where are the Agents ? Multi-Agent-Based Simulation II, Proceedings of MABS 2002, Third International Workshop. LNAI, Springer-Verlag.

Farges, J. L., Henry, J.J., Tufal, J. (1983). The Prodyn real-time traffic algorithm. The 4th IFAC Symposium on Transportation Systems, Baden Baden, Germany.

Fénies, P. (2006). Une méthodologie de modélisation par processus multiples et incrémentiels: application pour l'évaluation des performances de la Supply Chain. LIMOS. Clermont-Ferrand, Université Blaise Pascal.

Ferber, J. (1999). Multi-Agent Systems : An Introduction to Distributed Artificial Intelligence, Addison-Wesley Longman Publishing Co., Inc.

- Ferber, J. (2004). "The MadKit Project : a Multi-Agent Development Kit." from www.madkit.org
- Finder, N. V. and J. Strapp (1992). "A distributed approach to optimized control of street traffic signals." *Journal of Transportation Engineering* 118: 99-110.
- Furth, P.G. and T.H.J. Muller (2000). Conditional Bus Priority at Signalized Intersections: Better Service Quality with Less Traffic Disruption. *Transportation Research Record* 1731, pp. 23-30.
- Gartner, N. H. (1983). "OPAC: A demand-responsive strategy for traffic signal control." *Transportation Research Record* 906: 75–84.
- Goujon, J. Y. (1998). Un environnement de modélisation multi-agents pour la spécification et l'évaluation des performances des systèmes industriels de production. Thèse de doctorat en informatique. LIMOS. Clermont-Ferrand, Université Blaise Pascal.
- Gourgand, M., Kellert, P., (1991). Conception d'un environnement de modélisation des systèmes de production. troisième congrès de Génie Industriel. Tours.
- Grasso, B., M. Ward, et al. (1995). "ATMS and wide-area traffic management." *Proceedings of the Annual Meeting of the ITS America I(ITS America)*: 543–553.
- Gutknecht, O. (2001). Proposition d'un modèle organisationnel générique de systèmes multi-agents Examen de ses conséquences formelles, implémentatoires et méthodologiques. Sciences et Techniques du Languedoc. Montpellier, Université Montpellier II.
- Gutknecht, O. and J. Ferber (1999). "Vers une méthodologie organisationnelle de conception de systèmes multi-agents." *Unité mixte CNRS – Université Montpellier II C 09928*: 1-12.
- Hernandez, J., Ossowski, S., Garcia-Serrano, A., (2002). Multiagent architectures for intelligent traffic management systems. *Transportation Research*.
- Hostin, D. (2006). Conception et développement d'un simulateur de trafic urbain selon la méthode ASCImi-STU. LIMOS. Clermont-Ferrand, Université Blaise Pascal.
- Huhns, M. a. S., M. P. (2001). *Interaction-oriented programming*. AOSE-2000 Agent-Oriented Software Engineering, Berlin, Springer.
- Hunt, P. B., Robertson, D.L., Bretherton, R.D., Royle, M.C., (1982). "The SCOOT online traffic signal optimisation technique." *Traffic Engineering & Control* 23: 190-199.
- Iftar, A. (1997). "An intelligent control approach to decentralized routing and flow control in highways." *Proceedings of the 1997 IEEE International Symposium on Intelligent Control, Istanbul, Turkey*: 269–274.
- Lacomme P. (1998). Optimisation des systèmes de production : méthodes stochastiques et approches multiagents. LIMOS. Clermont-Ferrand, Université Blaise Pascal.

Li, M., J. Hallam, et al. (1996). "A cooperative intelligent system for urban traffic problems." Proceedings of the 1996 IEEE International Symposium on Intelligent Control, Dearborn, MI: 162–167.

Logi, F. (1999). CARTESIUS: A cooperative approach to real-time decision support for multi-jurisdictional traffic congestion management. Irvine, University of California. PhD Dissertation.

Lowrie, P. R. (1982). SCATS: The Sydney co-ordinated adaptive traffic system—principles, methodology, algorithms. Proceedings of the IEE International Conference on Road Traffic Signalling, London, UK.

Mansour, S. (2007). Un modèle de gestion distribuée de groupes ouverts et dynamiques d'agents mobiles Sciences et Techniques de Pau, Université de Pau et des Pays de l'Adour.

Mauro, V., Di Taranto, C. (1989). UTOPIA. IFAC/IFIP/IFORS on Control, Computers and Communications in Transport. Paris.

Michel, F. (2004). Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents. Sciences et Techniques du Languedoc. Montpellier, Université Montpellier II.

Mirchandani, P., Head, L. (1998). RHODES: A real-time traffic signal control system: Architecture, algorithms, and analysis. TRISTAN (Triennial Symposium on Transportation Analysis). San Juan, Puerto Rico.

Molina, M., F. Logi, et al. (1995). "An architecture integrating symbolic and connectionist models for traffic management center decision support." Proceedings of the fourth International Conference on Applications.

OECD (1990). Workshop on Knowledge-Based Expert Systems in Transportation Heikki Jams. Espoo, Finland.

OECD (1992). Workshop on Knowledge-Based Expert Systems in Transportation Montreal, Canada.

Ossowski, S., Cuenca, J., Garcia-Serrano, A., (1998). A case of multiagent decision support: using autonomous agents for urban traffic control. Iberoamerican Conference of Artificial Intelligence.

Ossowski, S., J. Z. Hernandez, et al. (2005). "Decision support for traffic management based on organisational and communicative multiagent abstraction." Transportation Research Part C 13: 272-298.

P99-071-1., S. N. X. (2001). Régulation du trafic routier par feux de circulation. Spécification du dialogue Standard Contrôleur. Partie 1 : spécifications DIASER (DIALOGUE Standard pour les Équipements de Régulation). France.

Robertson, D. I. (1986). Research on TRANSYT and SCOOT methods of signal coordination. ITE.

Robertson, D. L. and R. D. Bretherton (1990). Recent enhancements to SCOOT - SCOOT version 2.4. 3rd Int. Conf. on Road Traffic Control/IEE, London.

Russell, S. J., Norvig, Peter., (2003). Artificial Intelligence : A Modern Approach, Pearson Education.

Sarramia, D. (2002). ASCI-mi : une méthodologie de modélisation multiple et incrémentielle. sciences pour l'ingénieur. Clermont-Ferrand, Blaise Pascal. Doctorat.

Scemama, G. (1992). "Développement d'un Système à Base de Connaissance Historique pour la Gestion du Trafic." Recherche Transport Sécurité, R.T.S. 5(22).

SCEMAMA, G., BLAQUIERE, A. , OLIVERO, P. (2000). CLAIRE++ Observatory : a new tool to know and assess congestion on road network. 7th World Congress on ITS. Turin Italie.

Tchernev, N. (1997). La modélisation du processus logistique dans les systèmes flexibles de production. LIMOS. Clermont-Ferrand, Université Blaise Pascal.

van Katwijk, R., van Koningsbruggen, P., (2002). Coordination of traffic management instruments using agent technology. Transportation Research Part C.

van Lamsweerde, A. (2000). Requirements Engineering in the Year 2000: A Research Perspective. the 22nd International Conference on Software Engineering (ICSE'00).

van Lamsweerde, A. (2001). Goal-Oriented Requirements Engineering: A Guided Tour. the 5th IEEE International Symposium on Requirements Engineering (RE'01).

Wooldridge, M., Dunne, P. E., (2000). Optimistic and disjunctive agent design problems. Intelligent Agents, VII: Proceedings of the Seventh International Workshop on Agent Theories, Architectures and Languages, ATAL-2000, Berlin, Springer.

Wooldridge, M., WeifS, G., Ciancarini, P. (2002). Agent-Oriented Software Engineering II. 2nd International Workshop, AOSE-2001, Berlin, Springer.

Wooldridge, M. and N. R. Jennings (1995). "Intelligent agents : theory and practice." The Knowledge Engineering review 10(2): 115-152.

McGraw-Hill. "Encyclopedia of Science & Technology." from www.accessscience.com.

Annexe A : Définitions et terminologies

Adaptatif : Se dit d'un carrefour à feux ou d'un contrôleur de carrefour qui fonctionne en adaptativité.

Adaptativité : Cas particulier de microrégulation qui consiste à faire varier la durée d'un ou plusieurs phases de circulation ou d'une ou plusieurs lignes de diagramme, entre un minimum non nul et un maximum prédéfinis, en fonction de la demande de trafic. La méthode la plus couramment employée est celle de l'intervalle-véhicule.

Algorithme (ou méthode) de choix de plans de feux : Procédure logique suivant laquelle se fait la sélection d'un plan de feux, dans une bibliothèque de plans de feux préétablis (e.g. algorithme à seuil, algorithme du vecteur).

Appel : Signal extérieur au contrôleur caractérisant une demande particulière (e.g. appel bus, appel piéton).

Calendrier : Moyen de dater (horodatage) les événements.

Capteur : Dispositif physique sensible au passage ou à la présence de véhicules ou de piétons.

Carrefour : Zone de communication entre deux ou plusieurs routes, permettant aux véhicules le passage de l'une à l'autre. Les carrefours peuvent être à niveau (intersection) ou dénivelés.

Un carrefour est un ensemble de lignes de feux fonctionnant selon le même mode et sur la même base de temps.

Carrefour à feux : Intersection ou ensemble d'intersections dont les entrées et généralement les traversées piétonnes sont contrôlées par des signaux lumineux d'intersection. Par extension, le terme s'applique à une traversée piétonne en section courante contrôlée par feux, ou à la gestion par feux d'une section de voie sous alternat.

Carrefour composé : Ensemble de carrefours à feux élémentaires suffisamment rapprochés et imbriqués les uns dans les autres pour apparaître dans les yeux de l'utilisateur comme un seul carrefour et nécessiter des corrélations internes entre ligne de commande des feux. L'ensemble des courants est décomposable en au moins deux sous-ensembles tels que deux courants quelconques de deux sous-ensembles différents sont toujours compatibles. La matrice des incompatibilités d'un tel carrefour est décomposable. Dans un carrefour composé,

certaines mouvements ont à franchir plusieurs lignes de feux, mais à l'inverse, il existe des carrefours élémentaires où un véhicule peut également franchir plusieurs lignes de feux.

Carrefour élémentaire : Carrefour à feux dont la matrice des incompatibilités n'est pas décomposable.

Carrefour isolé : Carrefour à feux suffisamment éloigné des autres pour pouvoir y considérer l'arrivée des véhicules comme aléatoire.

Choix de plan de feux : Procédure de sélection d'un plan de feux, résultant soit de l'action d'un algorithme de choix de plan de feux par analyse du trafic en temps réel, soit d'une programmation horaire, soit d'un forçage manuel ...

Clignotant : Etat de fonctionnement d'un feu, jaune ou orange, où le feu est alternativement allumé et éteint à une fréquence voisine de 1 hertz.

Commande de coordination : Information permanente envoyée au contrôleur, lui ordonnant de fonctionner en mode coordonné.

Commutation de plans de feux : Passage d'un plan de feux à un autre.

Comptage : Evaluation du nombre de véhicules ou de piétons passant en un certain point pendant une période donnée. Le comptage peut être manuel ou automatique.

Comptage directionnel : Comptage réalisé à un carrefour, qui mesure le débit des différents mouvements directionnels pendant une période déterminée. Le plus souvent, on distingue les catégories de véhicules : deux-roues (2R), véhicules légers (VL) et poids-lourds (PL).

Concentration : Variable du trafic caractérisant le nombre d'u.v.p. ou de véhicules par unité de longueur de chaussée ou de voie de circulation à un instant donné, exprimée en général en véhicules par kilomètre.

Concentration critique : valeur de la concentration correspondant au débit maximal; en section courante (hors des files d'attente normales aux carrefours), c'est la limite entre fluidité et congestion.

Congestion : Remplissage excessif d'un réseau ou d'un tronçon qui résulte soit d'un blocage, soit d'une saturation. En théorie du trafic, la congestion qualifie un régime de circulation où la concentration est supérieure à la concentration critique.

Contrôleur de carrefour à feux (CAF) : C'est un automate qui assure la commande de l'ensemble des feux implantés sur un carrefour ainsi que tout ou partie de leur contrôle. Cette

commande de feux peut résulter d'une logique interne au contrôleur alimenté ou non par des données de trafic issues de capteurs ou d'une information reçue d'un organe central de commande, s'il existe. Le contrôleur assure également le contrôle du bon fonctionnement des feux d'un ou de plusieurs carrefours élémentaires et garantit en particulier le respect de contraintes de sécurité dans le déroulement des couleurs de feux. Les sécurités de fonctionnement assignées au contrôleur consistent à vérifier entre autre la conformité entre la commande issue du contrôleur et l'état d'allumage du feu. En cas de défaut majeur constaté, il peut assurer le passage de l'intersection en mode dégradé (jaune clignotant de sécurité noté JCS, voire extinction générale des feux notée EXG). Enfin, il distribue l'énergie électrique à partir du réseau d'alimentation EDF pour assurer l'allumage des sources lumineuses.

Un contrôleur asservi est un contrôleur dont la durée de certains des états est imposée par un organe extérieur comme par exemple un coordinateur de zone ou un poste central de régulation.

Coordinateur de zone : C'est un automate qui asservit plusieurs contrôleurs d'une zone donnée et assure une coordination des carrefours entre eux.

Coordination : Fonctionnement interdépendant de carrefours à feux dans le but de satisfaire un objectif global. Il s'agit le plus souvent d'un asservissement des carrefours à feux d'une zone ou d'un axe à une même référence de temps par l'application d'un cycle commun à tous les carrefours et de décalages constituant un plan de feux. Cette coordination permet de satisfaire un objectif déterminé comme par exemple la minimisation du nombre d'arrêts ou des temps de parcours.

Coordination **unitop** : coordination dans laquelle une seule référence est donnée par cycle, par une impulsion.

Coordination multi-impulsionnelle : coordination dans laquelle plusieurs ordres sont donnés à chaque contrôleur par cycle. Elle permet d'imposer entre autres de façon décentralisée la répartition de plusieurs durées d'états.

Cycle : Temps séparant deux passages successifs par le même état : le cycle est la période de temps après laquelle les mêmes événements se reproduisent dans le même ordre. La durée du cycle est la somme d'un temps de base et d'une dilatation éventuellement nulle..

Débit : Nombre de véhicules ou de piétons qui passent un en point donné par unité de temps sur une voie, sur un sens ou les deux sens de circulation d'une chaussée, ou sur un sens ou les deux sens d'un passage piéton. Pour les véhicules, il peut être exprimé par exemple en

véhicules par heure, en u.v.p./h, ou pour une entrée de carrefour à feux, en u.v.p.d. par heure de feu vert en considérant le vert permanent.

Débit de saturation d'une entrée de carrefour : débit maximal de l'entrée par heure de feu vert ou en supposant son vert permanent.

Débit de pointe : débit maximal de véhicules qui passent effectivement sur une section de route pendant un nombre donné de minutes consécutives (e.g. débit du quart de pointe); généralement exprimé en u.v.p./h.

Détecteur : Dispositif qui met en forme les informations issues du (ou des) capteur(s).

Diagramme des feux : Représentation graphique de la chronologie de fonctionnement de l'ensemble des lignes de feux d'une signalisation d'intersection (cf exemple dans définition du diagramme séquentiel).

Diagramme des phases : Diagramme des feux dans lequel sont représentées (cf figure A-1 ci-dessous) toutes les phases et éventuellement certains autre états importants du déroulement des feux.

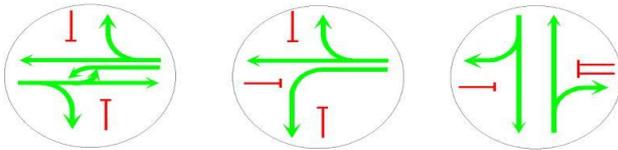


Figure A-1 : exemple pour un cycle en trois phases.

Diagramme séquentiel : Diagramme des feux dans lequel est représentée une succession d'états dont chacun débute et se termine par la modification d'au moins un signal (cf figure 4-9). Dans ce type de diagramme, certains états prennent en compte de façon préétablie les contraintes de temps de sécurité

Escamotage

Type de microrégulation qui consiste à supprimer tout ou partie d'une ou plusieurs lignes en l'absence de demande sur l'entrée ou les entrées véhicules ou piétonnes correspondantes.

L'escamotage d'état ou de phase se traduit par une amputation de la durée du cycle égale à la durée de l'état ou de la phase escamoté, ou bien par un report de cette durée sur un autre état si la durée du cycle est imposé.

L'escamotage de ligne consiste dans le maintien au rouge d'une ligne avec éventuellement maintien au vert d'au moins une autre ligne sans qu'il y ait nécessairement un effet sur le déroulement des autres lignes de diagramme indépendantes.

Feu de circulation

Les feux lumineux réglementant la circulation des véhicules ou la traversée des piétons sont verts, jaunes ou rouges. Ils sont blancs lorsqu'ils ne concernent que les tramways. Eventuellement, ils sont dotés d'un pictogramme désignant la catégorie d'usagers auxquels ils s'adressent. Les couleurs jaune, rouge et blanc peuvent éventuellement clignoter.

Fonctionnement cyclique

Fonctionnement d'un carrefour à feux qui peut être décrit par une seule séquence d'états prédéfinis de l'ensemble des feux du carrefour.

Fonctionnement à temps fixes : fonctionnement cyclique dans lequel la durée de tous les états est fixe.

Fonctionnement à cycle fixe : fonctionnement cyclique dans lequel la durée du cycle est constante, soit parce que toutes les durées des différents états sont constantes, soit, en général pour assurer une coordination, parce que la durée du cycle est imposée (les variations éventuelles de durées sont, dans ce dernier cas, compensées par des reports de vert à l'intérieur du cycle; la durée du cycle s'observe alors entre deux passages successifs à un point de référence – voir recalage).

Fonctionnement à cycle variable : fonctionnement cyclique dont la durée du cycle n'est pas imposée et où l'occurrence et la durée des états peut varier aussi bien par microrégulation, que par macrorégulation. La durée du cycle s'observe entre deux passages successifs des feux par un point de référence, à la condition qu'entre les deux, tous les états soient apparus; cette durée varie en général d'un cycle à l'autre.

Fonctionnement séquentiel

Dans un fonctionnement cyclique, façon de décrire le déroulement des lignes par un enchaînement d'états de l'ensemble des feux, dont la nature et l'ordre sont figés, mais dont la durée, éventuellement nulle, peut varier par adaptativité et escamotage. Ce type de fonctionnement est facilement représenté par un diagramme séquentiel.

Forçage manuel

Action manuelle imposant l'application d'un plan de feux en abandonnant l'action de sélection et de commutation automatique

Glissement de lignes

Cas d'adaptativité qui consiste à faire varier la durée des verts de plusieurs lignes de feux admises dans la même phase, indépendamment les unes des autres, en général lorsqu'on peut utiliser les réductions de vert éventuelles en les reportant sur certaines lignes au vert dans la phase suivante. Le glissement ne peut pas être représenté sous la forme d'un diagramme totalement séquentiel.

Groupe de signaux

Ensemble de signaux qui gèrent un même courant de circulation. Pour les signaux tricolores, il comprend les signaux principaux et leurs signaux répéteurs. Le fonctionnement d'un groupe de signaux est représenté par une ligne de diagramme.

Intersection

Lieu de jonction ou de croisement à niveau de plusieurs chaussées, quels que soient les angles des axes de ces chaussées.

Intervalle critique

Valeur de seuil d'intervalle-véhicule à partir duquel on considère que la circulation ne s'effectue plus en pelotons. C'est la valeur préprogrammée dans la méthode d'adaptativité par intervalle-véhicule à partir de laquelle on estime inutile de prolonger le vert.

Intervalle-véhicule

Intervalle de temps entre deux véhicules successifs détectés sur un capteur. Par extension, méthode d'adaptativité qui consiste à prolonger le vert tant que l'intervalle-véhicule est inférieur à un seuil appelé intervalle critique. Pour réaliser cette fonction, l'intervalle – véhicule à utiliser est le créneau, c'est à dire l'intervalle de temps entre l'arrière d'un véhicule et l'avant du véhicule qui le suit, ou encore l'intervalle de temps entre le front de descente du signal du premier véhicule et le front de montée du signal du véhicule suivant.

Ligne d'effet des feux

Ligne au droit de laquelle les véhicules doivent se soumettre aux indications des signaux. Cette ligne est en général matérialisée par un marquage approprié portant le même nom ou par

le marquage d'un passage pour piétons; à défaut de marquage, elle se situe dans un plan passant par le signal et perpendiculaire à l'axe de la voie.

Ligne de diagramme

Dans un diagramme de feux, représentation graphique de la suite des états d'un seul groupe de signaux (ou éventuellement de plusieurs groupes synchrones), pendant un cycle. Elle permet de connaître les instants d'allumage et d'extinction de chacune des couleurs des signaux de ce groupe, relativement à l'état des feux des autres lignes du diagramme.

Ligne de commande de feux ou ligne de feux

Dispositif d'allumage des feux conformément à une ligne de diagramme; il en existe 5 types : tricolore vert – jaune - rouge, tricolore jaune clignotant sur le feu du bas – jaune fixe – rouge, tramway barre verticale – disque – barre horizontale, piétonne et d'anticipation.

En d'autres termes, une ligne de feux est un ensemble de signaux recevant des indications d'allumage identique des feux pour un même courant de circulation.

Lignes de feux antagonistes : lignes de feux qui gèrent des courants de véhicules ou de piétons incompatibles entre eux.

Macrorégulation

En définissant une stratégie commune à un ensemble de carrefours et en imposant un fonctionnement dicté par des considérations générales, on met en place un réglage dit de macrorégulation. En général, ce rôle est confié à un poste central, celui-ci opère à distance sur les contrôleurs via des coordonnateurs de zone. Pour que ce type de régulation puisse s'appliquer, un certain nombre de données de fonctionnement doivent être paramétrées comme les cycles.

Matrice des incompatibilités (ou des antagonismes)

Matrice carrée symétrique de dimension N (N = nombre de courants de circulation de véhicules ou de piétons dans un carrefour) dans laquelle chaque élément binaire indique si les deux courants auxquels il correspond peuvent être admis simultanément ou pas.

Microrégulation

Action de régulation qui permet de modifier l'occurrence, l'ordre, la durée des phases trafic ou des lignes de feux, à partir de l'analyse instantanée de la demande «Véhicules» ou

«Piétons», sur un seul carrefour. L'adaptativité, l'escamotage, le glissement, l'aiguillage et le repos sont des formes de microrégulation.

Mode asservi

Mode de fonctionnement d'un contrôleur de carrefour dans lequel la durée de certains états et éventuellement la durée ou l'apparition de certaines phases ou lignes de diagramme sont assujetties à une commande extérieure (voir coordination).

Mode de fonctionnement

Type de fonctionnement d'un contrôleur de carrefour à feux; on peut distinguer le mode clignotant, le mode tricolore, le mode extinction, etc ...

Mode local

Mode de fonctionnement d'un contrôleur de carrefour, déterminé exclusivement par sa programmation propre et les détecteurs qui lui sont associés, indépendamment de toute commande extérieure.

Mode manuel

Mode de fonctionnement séquentiel à temps fixe dont la durée de certains états est contrôlée par l'action sur le bouton-poussoir de commande manuelle.

Mode tricolore

Mode de fonctionnement dans lequel les couleurs des signaux se déroulent selon les règles normales pour l'utilisateur. On parle également de mode normal. Le mode tricolore peut être local, manuel, asservi.

Multiprogrammation

Technique de commande des signaux d'un ou plusieurs carrefours à feux qui consiste à disposer de plusieurs plans de feux correspondant aux situations types prévisibles du trafic et à les mettre en œuvre selon un calendrier programmé ou par analyse du trafic. Ces différents plans de feux peuvent être programmés dans chaque contrôleur ou résulter d'une commande extérieure (mode asservi).

Paramètres de choix de plans de feux

Données propres à chaque site, nécessaires au fonctionnement de l'algorithme de choix de plan de feux. Exemple : débit, taux d'occupation, calendrier, etc ...

Phasage

Découpage (ou ordonnancement) en phases de circulation du fonctionnement d'un carrefour à feux.

Phase de circulation ou de trafic

Dans la description du fonctionnement d'un carrefour à feux, période pendant laquelle une ou plusieurs lignes de feux sont admises au vert simultanément. Entre deux phases s'écoulent obligatoirement les temps de sécurité (jaune et rouge de dégagement).

Plan de feux

Ensemble de données définissant tout ou partie du fonctionnement d'un carrefour à feux (ou d'un ensemble).

Les paramètres de structure d'un plan de feux définissent notamment la nature et l'enchaînement des états de l'ensemble des lignes de feux.

Les paramètres de durée d'un plan de feux définissent les durées minimales et maximales.

Plan de feux de base

Dans un fonctionnement cyclique, plan de feux issu du plan de feux de sécurité qui décrit la structure du diagramme, impose la durée des états invariables et une durée pour les autres. Le plan de feux de base peut être le plan appliqué en mode local en cas de dysfonctionnement de certains éléments propres aux plans de feux de fonctionnement (e.g. panne de détecteur, rupture de liaison d'asservissement).

Plan de feux de fonctionnement

Ensemble des données définissant complètement le déroulement des états de toutes les lignes de commande des signaux d'un carrefour à feux.

Pour un carrefour isolé, les données principales du plan de feux déterminent la nature et la durée de tous les états.

Pour un carrefour coordonné, le plan de feux définit, en outre, le décalage du carrefour par rapport à une référence de temps.

Dans un fonctionnement cyclique, les plans de feux de fonctionnement à même structure découlent d'un même plan de feux de base.

Plan de feux de sécurité

Dans un fonctionnement cyclique, plan de feux qui satisfait aux contraintes de temps de sécurité minimaux : rouge de dégagement, jaunes fixes et verts minimaux, et dont les durées sont incompressibles. Ce plan de feux n'est généralement pas applicable sur le terrain (e.g. durée de cycle trop courte); cependant, il sert à définir les contraintes de sécurité imposées à tous les plans de feux.

Recalage

Action de remise à l'heure d'une horloge locale. Par exemple, à un signal électrique donné, l'horloge interne d'un contrôleur se positionne à une heure préprogrammée.

Répétiteur

Signal de circulation donnant une information utile à l'utilisateur répétant ou complétant la signalisation principale sans avoir de caractère prescriptif; il peut s'agir d'un signal de dimension réduite, situé à faible hauteur, qui répète rigoureusement les indications du signal principal situé sur le même support.

Repos

Principe d'action d'une microrégulation dans laquelle le déroulement du diagramme est arrêté sur un point d'arrêt dit position de repos, en attente d'appel.

Repos sur artère : repos sur une phase présélectionnée.

Repos sur dernier appel : repos sur la dernière phase appelée.

Repos au rouge intégral : repos sur un état de rouge intégral.

Rouge de dégagement

Période minimale qui s'écoule entre le passage au rouge d'une ligne de feux et l'ouverture d'une ligne de feux qui est déclarée antagoniste à la première. Sa durée est calculée pour permettre aux usagers engagés à la dernière seconde de vert véhicule ou piétons ou de jaune fixe véhicule de la première ligne de feux, de dégager la zone des conflits avant l'arrivée des véhicules ou piétons (cf figure A-2 ci-dessous) s'engageant à l'ouverture de la deuxième.

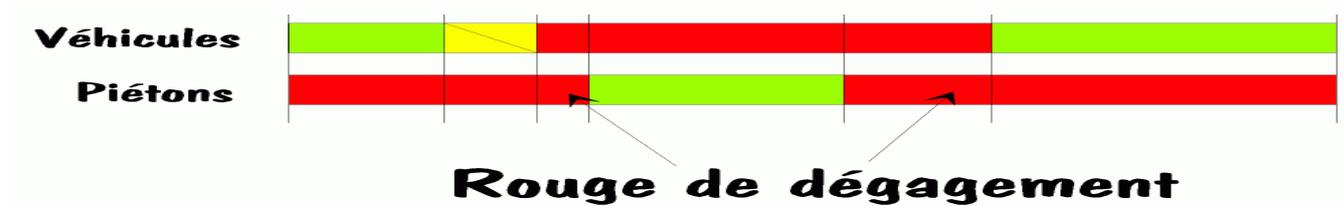


Figure A-2 : illustration des durées de rouge de dégagement pour deux lignes véhicules et piétons en conflit direct.

A chaque couple de lignes de feux antagonistes A et B, sont associées une durée de rouge de dégagement entre A et B, et une autre entre B et A.

Pour les tramways, il est préférable d'avoir recours à une détection permettant de s'assurer que ce véhicule a dégagé la zone de conflit avant d'engager la phase suivante. A défaut, il convient de tenir compte des caractéristiques propres aux tramways (vitesse, freinage, longueur...).

Rouge intégral

Etat d'un carrefour lorsque tous les feux rouges destinés aux véhicules ou aux piétons sont allumés, les signaux d'anticipation étant éteints. Cet état se présente lors d'un repos au rouge intégral.

Saturation

Etat de la circulation lorsque la demande dépasse l'offre; une entrée de carrefour à feux est saturée lorsque la file d'attente au rouge n'est pas complètement résorbée à l'issue du premier temps de vert; un carrefour à feux est saturé si l'une au moins de ses entrées est saturée.

En théorie du trafic, saturation est synonyme de congestion.

Signal d'anticipation

Signal composé d'un seul feu jaune clignotant accordant une autorisation de passage, en fin de rouge du signal tricolore circulaire auquel il est associé, à un mouvement directionnel (signal d'anticipation directionnel) ou à une catégorie de véhicules (signal d'anticipation modal) désigné par un pictogramme.

Signal principal

Signal lumineux de diamètre 200 ou 300 mm pour véhicules, destiné à être vu à distance pour permettre l'arrêt des véhicules dans des conditions normales et situé sur trottoir, îlot, accotement ou au-dessus de la chaussée. Les signaux de rappel sont des signaux principaux.

Signaux lumineux d'intersection

Signaux lumineux de circulation de circulation appartenant à une signalisation d'intersection. Les différents signaux d'intersection comprennent des signaux tricolores destinés aux véhicules, des signaux bicolores destinés aux piétons et des feux jaunes clignotants autorisant le passage anticipé d'un mouvement.

Station locale

Ensemble d'équipements de terrain faisant l'objet d'une même connexion à un système externe donné. Cet ensemble exécute une ou plusieurs applications telles que commande et contrôle de carrefours à feux élémentaires, mesure du trafic, affichage de messages variables, télésurveillance de la station locale, détection de véhicules de transport en commun,...

Système externe

Ensemble des moyens de régulation, de surveillance et de communication tels qu'ils peuvent être vus d'une station locale. Typiquement, un poste central d'exploitation ou un coordinateur de zone est un système externe vis à vis d'un contrôleur (qui est la station locale).

VTC

Acronyme de véhicule de transport en commun

Annexe B : Glossaire des acronymes

ASCI-mi : Analyse, Spécification, Conception, Implantation – multiple, incrémentielle : une méthodologie de modélisation développée dans l'équipe modélisation et aide à la décision du LIMOS

CAF : L'application Carrefour A Feux

CDD : Centre De Décision, ou Entité de Régulation (ER), est un contrôleur de carrefours à feux

CDZ : Coordinateur De Zone, coordonne un ensemble de contrôleurs

CTRL : Contrôleur, gère plusieurs carrefours à feux

ER : Entité de Régulation, voir CDD

MaSE : Multi-agent System Engineering, une méthode d'analyse et de conception basée sur les systèmes multi-agents

MDT : L'application Mesure De Trafic

PC : Poste Central, coordonne l'ensemble des entités de régulation

PMV : L'application Panneaux à Messages Variables

SAD : Système d'Aide à la Décision

SAE : Système temps réel d'Aide à l'Exploitation

SMA : Systèmes Multi-Agents

STU : Systèmes du Trafic Urbain

TAD : Tourner A Droit

TAG : Tourner A Gauche

TD : Tout Droit

VTC : L'application Véhicules de Transport en Commun

Modes de fonctionnement des STU:

ASS : mode ASServi, est un mode tricolore automatique

AUTO : mode AUTOMatique, est un mode tricolore

EXG : mode EXtinction Générale

INI : mode INItialisation

JCG : mode Jaune Clignotant Général

LOC : mode LOCal, est un mode tricolore automatique

MAN : mode MANuel, est un mode tricolore

TRI : Mode TRicolore

Annexe C : Des stratégies pour donner la priorité aux VTC :

L'idée est de considérer l'attente totale par personne et ne pas par véhicule. La stratégie générale est d'augmenter la durée de vert ou réduire la durée de rouge. Nous présentons deux grandes stratégies : Passive et Active (dynamique). Ensuite nous présentons la stratégie du contrôleur PRIBUSS (Davol A., 2001).

La stratégie passive :

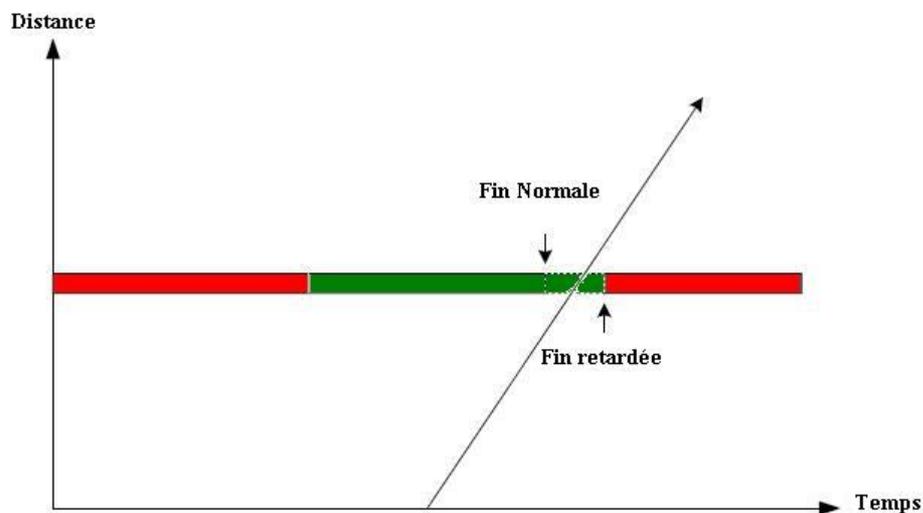
- a. Augmenter la durée de vert
- b. Faire la durée de phase plus courte (point négative : augmenter la pourcentage du temps perdues (temps de sécurité : 3 secondes de « toutes rouge ») et cela est indépendant de la durée du cycle).
- c. Diviser la phase de vert en 2 parties dans le même cycle.
- d. Coordination : onde vert correspondant à la vitesse moyen de VTC

Point positive : On n'a pas besoin de changer les matériels

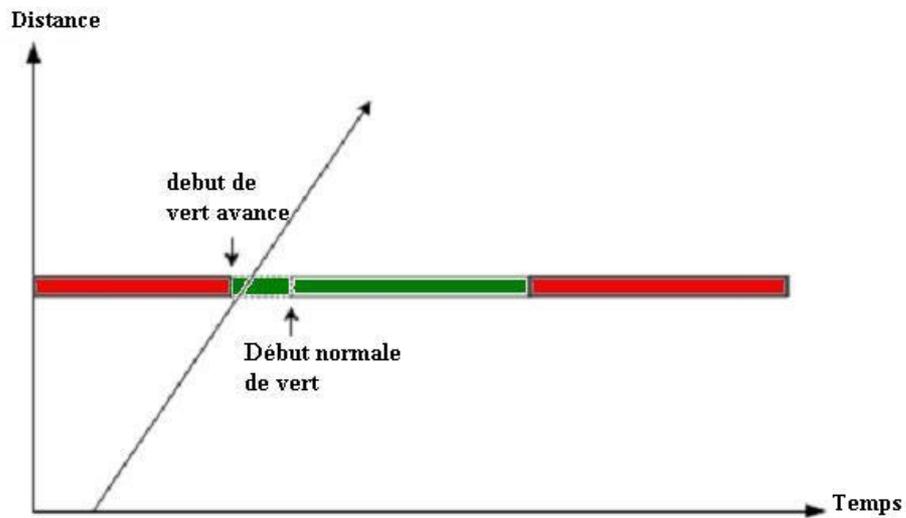
Point négative : avec une fréquence petite du passage de VTC, l'efficacité est moins

La stratégie active - dynamique

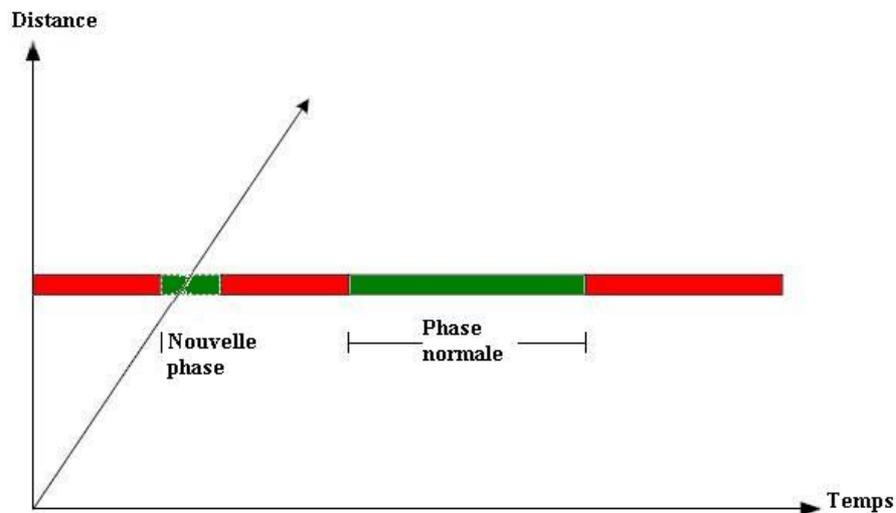
- a. Augmenter la durée de vert lorsque la VTC arrive vers la fin de la phase vert.



- b. Mettre fin de la phase rouge lorsque la VTC arrive vers la fin de la phase rouge.



- c. Insérer une nouvelle phase (vert) uniquement pour la VTC lorsqu'elle arrive dans la première moitié de la phase rouge.



- d. Recommencer la phase de vert :

Cette stratégie marche bien dans la situation normale, mais elle ne marche pas bien dans les situations de saturation.

Pour la stratégie active il y a **trois catégories** :

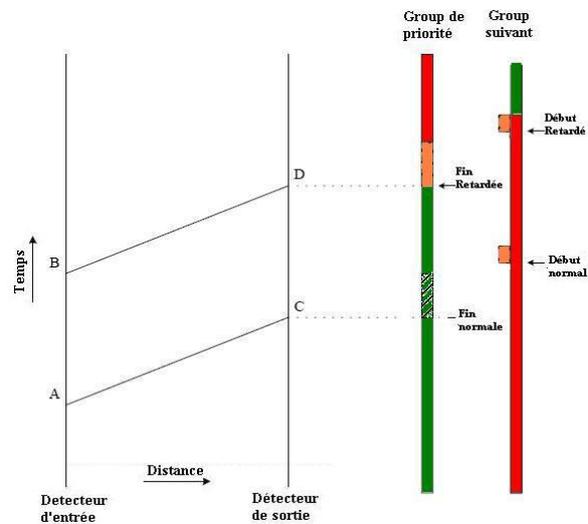
- 1- **Inconditionnée** : elle dépend d'une seule information c'est l'arrivée de VTC.
- 2- **Conditionnée** : elle prend en compte d'autres critères (le temps perdu aux arrêtes,...) [Furth et Muller, 2000]
- 3- **Adaptive** : balancer entre les VTC et les autres véhicules, on considère que un bus est égal à 50 véhicules

Difficultés: La stratégie adaptative essaye d'améliorer l'écoulement global du système (à partir des informations macroscopiques (elle ne compte pas le temps perdu aux arrêtes)), donc elle ne peut pas résoudre le problème d'un VTC, le problème de priorité de VTC est local et c'est au contrôleur de le résoudre (informations microscopiques)

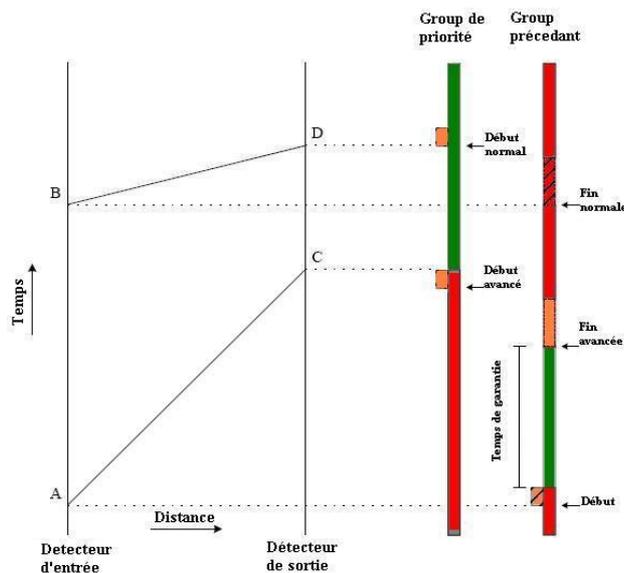
PRIBUSS

PRIBUSS est un contrôleur appliquant des stratégies actives pour donner la priorité au VTC, il nécessite deux détecteurs (entrée et sortie), quatre stratégies sont appliquées :

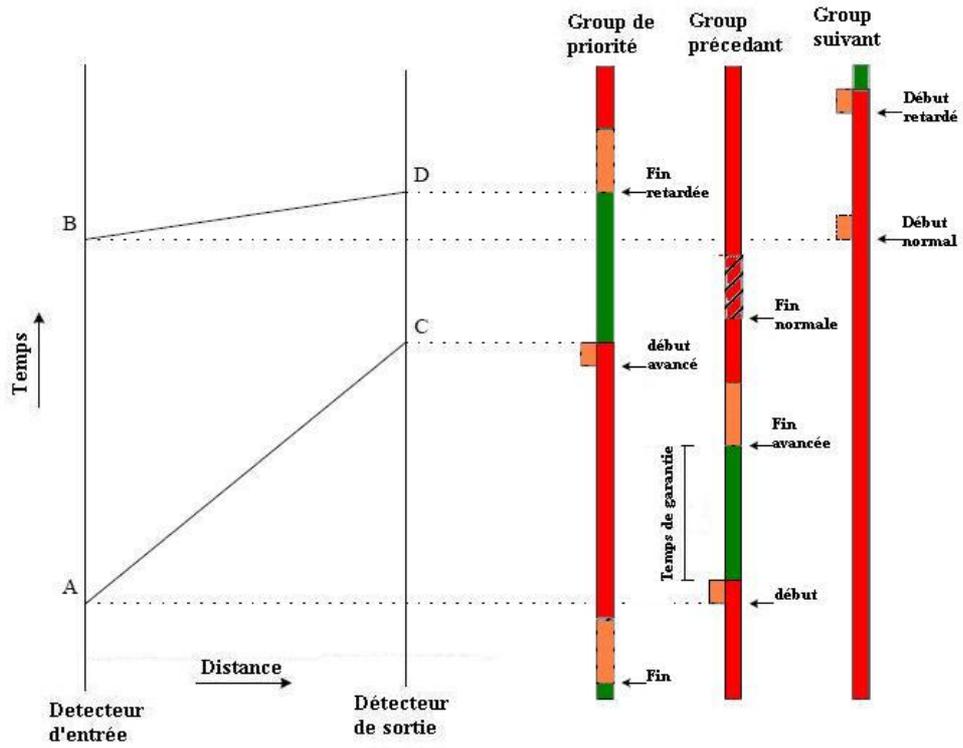
1- Augmenter la durée de vert



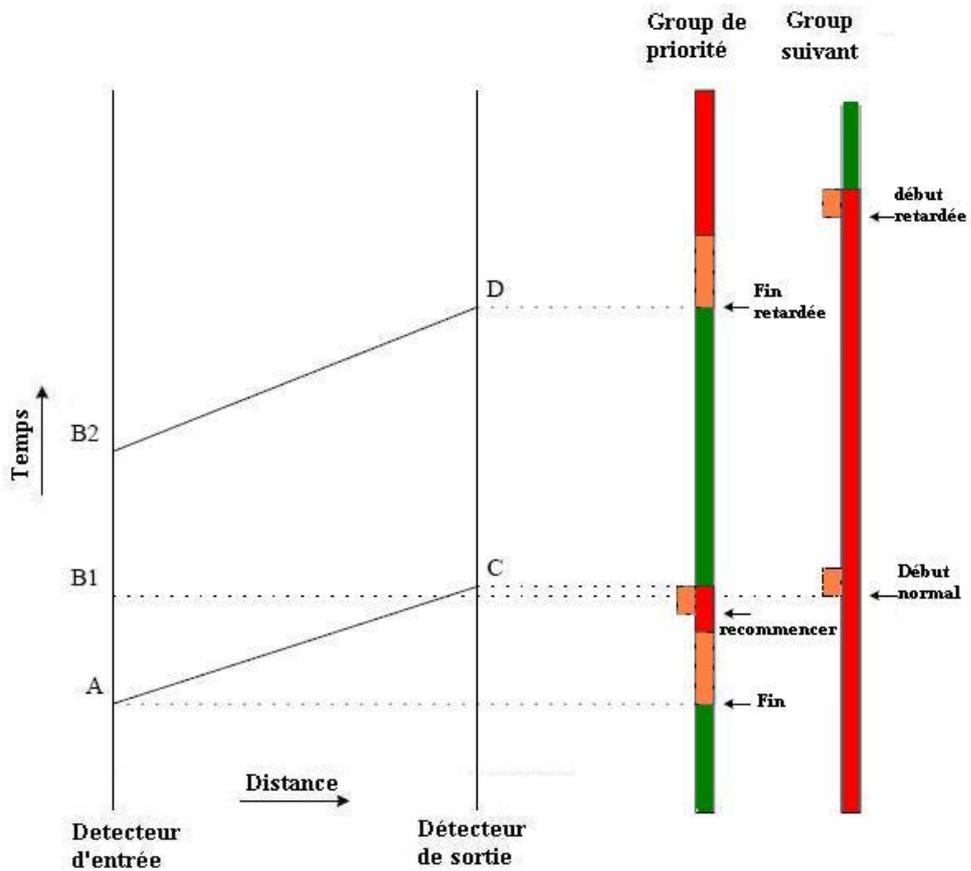
2- Mettre fin de la phase rouge



3- Insérer une nouvelle phase de vert



4- Recommencer la phase de vert



Annexe D : Diagrammes séquentielles des plans de feux sur les carrefours du Riom

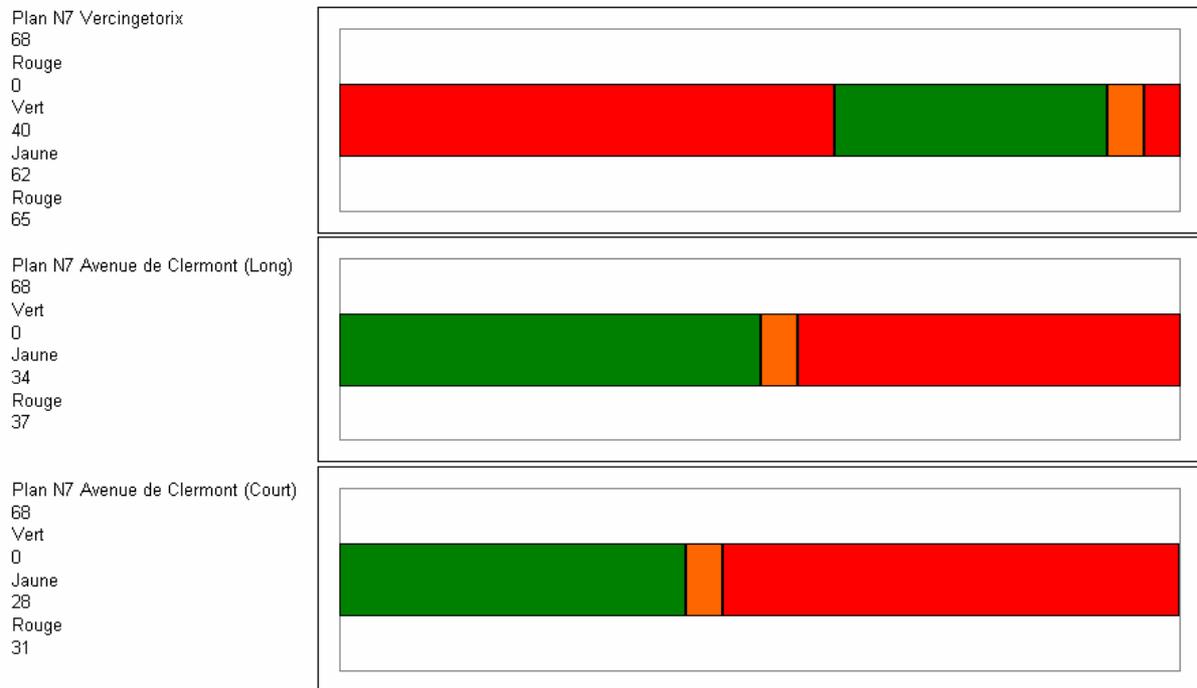


Figure C-1 : Carrefour N7

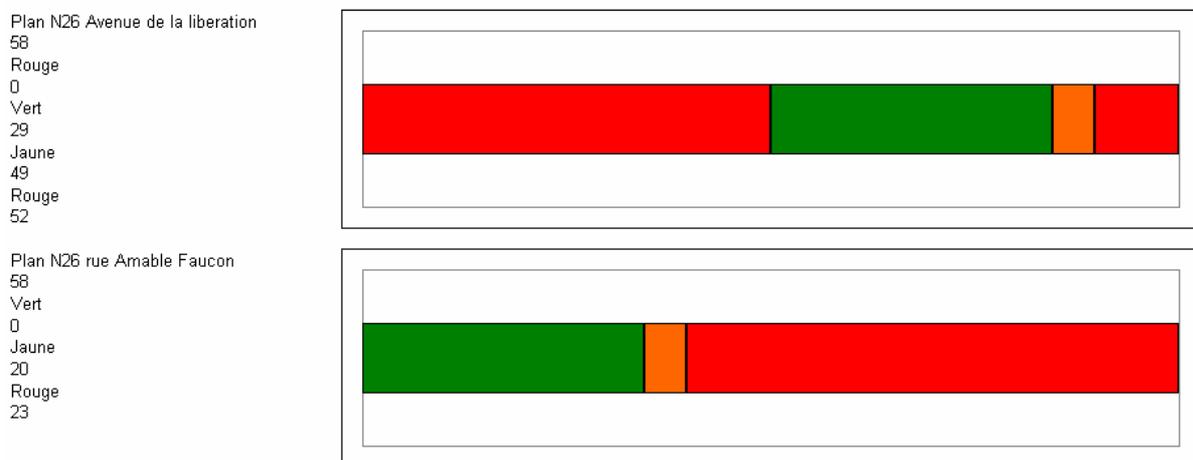
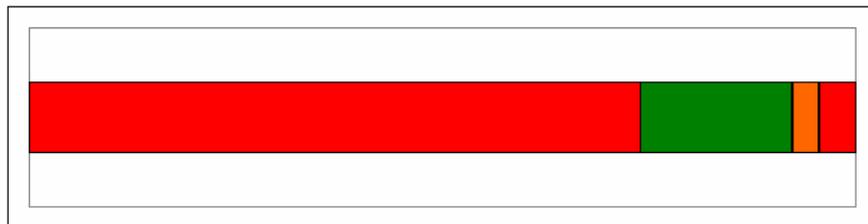


Figure C-2 : Carrefour N26

Plan NVert0
65
Rouge
0
Vert
48
Jaune
60
Rouge
62

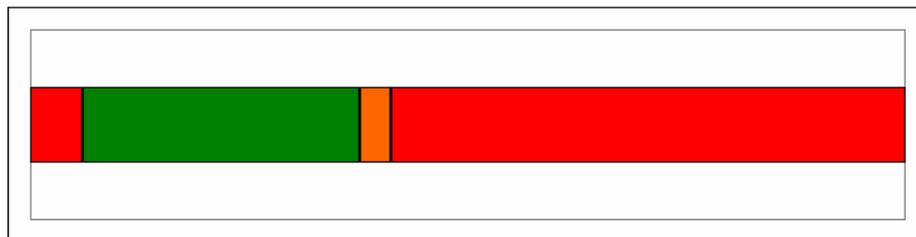


Plan NVert0 RN9
65
Vert
0
Jaune
40
Rouge
43



Figure C-3 : Carrefour N100

Plan NVert5 Route de Volvic
85
Rouge
0
Vert
5
Jaune
32
Rouge
35

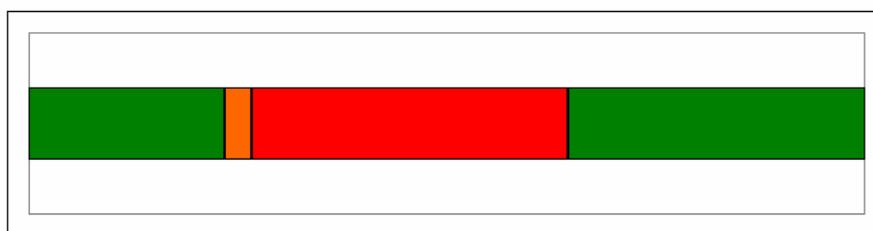


Plan NVert5 RN9
85
Rouge
0
Vert
40
Jaune
82



Figure C-4 : Carrefour N105

PLAN 2
90
Vert
0
Jaune
21
Rouge
24
Vert
58



PLAN 1
90
3
0
Vert
29
Jaune
50
Rouge
53



Figure C-5 : Carrefours N104, N106, N107, N108, N47

TITLE:

The multi-agents approach for the management of complex systems
Applied to urban traffic systems

ABSTRACT:

The design, the study and the management of urban traffic systems (UTS) are difficult and expensive and the construction and are very complex tasks. In these systems the goal is to improve traffic and optimize the flow through operating rules and the control system (which constitute the decisional subsystem).

The distributed approach of controlling systems can contribute to the development of agile organizations, combining the logic of network and hierarchy. However, the non-centralized control system of UTS fundamentally relies on the interactions of communicating, autonomous and capable of making decisions entities.

This work fits in the development of the modeling methodology ASDImi-UTS (analysis, specification, design, implementation - multiple, incremental for UTS) which aims to assist in the design and management of the UTS.

The problems involved in the design of the decisional subsystem of UTS are the control of spatial and temporal consistency of decisions, considering the characteristics of control systems and their environment. We propose for the decisional subsystem a multi-agent approach and a decision-making tool to improve the quality of service to users and to solve planning issues at the strategic, tactical and operational levels.

KEYWORDS:

Modeling methodology, Decision making aid, multiagents approach, Traffic control system, Traffic management, Decisional subsystem, Decisional model

TITRE :

L'approche multi-agents pour le pilotage des systèmes complexes
Appliquée aux systèmes du trafic urbain

RÉSUMÉ :

La conception, l'étude et la gestion des systèmes du trafic urbain (STU) sont difficiles et coûteuses et sont des tâches très complexes. Dans ces systèmes, l'objectif principal est d'améliorer la circulation et d'optimiser la fluidité grâce à des règles de fonctionnement et de pilotage (qui constituent le sous-système décisionnel).

L'approche distribuée du pilotage peut contribuer à la mise en place d'organisations agiles, combinant des logiques de réseaux et de hiérarchie. Cependant, le pilotage non centralisé des STU s'appuie fondamentalement sur les interactions d'entités communicantes, autonomes et capables de prendre des décisions.

Ce travail s'intègre dans le développement de la méthodologie de modélisation ASCImi-STU (analyse, spécification, conception, implantation – multiple, incrémentielle pour les STU) qui a pour objectif d'apporter une aide à la conception et à la gestion des STU.

La problématique liée à la conception du sous-système décisionnel des STU est la maîtrise de la cohérence spatiale et temporelle des décisions, compte tenu des caractéristiques des systèmes pilotés et de leur environnement. Nous proposons pour le sous-système décisionnel une approche multi-agents et un outil d'aide à la décision pour améliorer la qualité de service aux usagers et pour résoudre des problèmes de planification au niveau stratégique, tactique et opérationnel.

MOTS-CLÉS :

Méthodologie de modélisation, Systèmes d'aide à la décision, L'approche multi-agents, Systèmes d'aide à l'exploitation, Pilotage de trafic urbain, Sous-système décisionnel, Modèle décisionnel