



HAL
open science

Conception d'un système d'information pour l'aide au déplacement multimodal: Une approche multi-agents pour la recherche et la composition des itinéraires en ligne.

Mohamed Amine Kamoun

► To cite this version:

Mohamed Amine Kamoun. Conception d'un système d'information pour l'aide au déplacement multimodal: Une approche multi-agents pour la recherche et la composition des itinéraires en ligne.. Automatique / Robotique. Ecole Centrale de Lille, 2007. Français. NNT: . tel-00142340v2

HAL Id: tel-00142340

<https://theses.hal.science/tel-00142340v2>

Submitted on 23 Apr 2007

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.

N° d'ordre : 44

Ecole Centrale de Lille
Université des Sciences et Technologies de Lille

THESE

présentée en vue d'obtenir le grade de

DOCTEUR

Spécialité : Automatique et Informatique Industrielle

par

Mohamed Amine KAMOUN

Doctorat délivré conjointement par l'Université des Sciences et
Technologies de Lille et l'Ecole Centrale de Lille

**Conception d'un système d'information pour l'aide au déplacement multimodal :
Une approche multi-agents pour la recherche et la composition des itinéraires en ligne.**

Soutenue publiquement le 4 Avril 2007 devant le jury :

Président :	Etienne CRAYE	Professeur, Ecole Centrale de Lille
Rapporteurs :	Jacques CARLIER	Professeur, Université de Technologie de Compiègne
	Abdallah ELMOUDNI	Professeur, Université UTBM de Belfort-Montbéliard
Examineurs :	Arnaud FREVILLE	Professeur, Direction Recherche du CR-NPDC
	René MANDIAU	Professeur, Université de Valenciennes
Directeur :	Slim HAMMADI	Professeur, Ecole Centrale de Lille
Co-Directeur :	Guillaume USTER	Chargé de Recherche, INRETS

Thèse préparée dans les laboratoires : LAGIS à l'Ecole Centrale de Lille et l'Université des Sciences et Technologies de Lille / ESTAS de l'INRETS.

A mes parents et ma tante Yaya,
à ma femme Sana et ma fille Eya,
à ma soeur et mon frère.
A toute ma grande famille.

Remerciements

Cette thèse est l'aboutissement de travaux élaborés dans le cadre d'une coopération entre l'équipe ESTAS "Evaluation des Systèmes de Transport Automatisés et de leur Sécurité" de l'INRETS "Institut National de Recherche sur les Transports et leur Sécurité", et l'équipe ID "Ingénierie de Décision" du laboratoire LAGIS, "Laboratoire d'Automatique Génie Informatique et de Signal".

Je tiens d'abord à exprimer ma profonde reconnaissance au Professeur Slim HAMMADI, Professeur à l'Ecole Centrale de Lille et directeur de cette thèse, pour l'énorme soutien moral et scientifique qu'il m'a généreusement accordé tout au long de ces trois années. Je le remercie aussi pour ses grandes qualités humaines.

Je tiens ensuite à remercier le Docteur Guillaume USTER, Chargé de recherche à l'INRETS, co-directeur de cette thèse, et expert en information multimodale, de m'avoir fait découvrir ce domaine de recherche fort intéressant. Je le remercie aussi pour ses précieux conseils, et pour m'avoir toujours encouragé à aller de l'avant.

Je ne saurais oublier de remercier, le Professeur Pierre BORNE, qui a accompagné mes premiers pas à Lille, et qui m'a accueilli depuis mon projet de fin d'étude ingénieur, au sein de son équipe ID, à l'Ecole Centrale de Lille.

Mes sincères remerciements vont aussi au Professeur Etienne CRAYE, Professeur à l'Ecole Centrale de Lille et Directeur de cet honorable établissement, pour l'honneur qu'il m'a fait en acceptant de présider ce jury.

Il m'est aussi agréable d'exprimer toute ma reconnaissance aux :

- Professeur Abdallah ELMOUDNI, Professeur à l'Université de Technologie Belfort-Montbéliard,
- Professeur Jacques CARLIER, Professeur à l'Université de Technologie de Compiègne, de m'avoir fait l'honneur en s'intéressant à mon travail de thèse, et en acceptant de le rapporter.

Je tiens également à remercier :

- le professeur Arnaud FREVILLE, Directeur à la Direction Recherche, Enseignement Supérieur, Santé, Technologies de l'Information et de la Communication au Conseil Régional Nord-Pas De Calais,
 - le Professeur René Mandiau, Professeur à l'Université de Valenciennes et du Hainaux-Cambrésis,
- d'avoir accepté d'examiner ma thèse et de faire partie de mon jury.

Je souhaite aussi témoigner ma gratitude à mes collègues d'ESTAS, et à ceux du LAGIS, en particulier Fabien DEGOUTIN, Othman LAHLOU, Jérôme ROCHETEAU, Christelle Petit-Roze, Kamel ZIDI, et Mohamed Mahmoud OULD SIDI, pour tous les moments de bonheur et de labeur qu'on a passés ensemble.

Enfin, une pensée très particulière et une reconnaissance infinie :

- A mes parents, ma tante Yaya, ma femme Sana, ma fille Eya, ma soeur et mon frère, pour leur amour, leur patience et leurs sacrifices.
- Aux familles KSONTINI et DAMAK qui m'ont généreusement accueilli à Lille.

Table des matières

Introduction générale	15
1 Systèmes d'Information Multimodale	19
1.1 Introduction	19
1.2 Définition de l'information multimodale	20
1.3 Gestion des réseaux de transport	21
1.3.1 Contexte et organisation des transports publics en France	21
1.3.2 Contexte et organisation des transports publics en Europe	23
1.3.3 Configuration d'un réseau multimodal multi opérateur	23
1.4 Systèmes d'Information pour le transport	25
1.4.1 Définition d'un Système d'information	26
1.4.2 Cartographie de l'information multimodale à l'image de la configura- tion des réseaux	26
1.4.3 SIM : spécification des besoins	27
1.4.4 Freins à la production de l'information multimodale	28
1.5 Intégration des systèmes d'information	29
1.5.1 Veille technologique sur l'intégration de l'information : EAI	29
1.5.2 Intégration de l'information voyageur : État de l'art	34
1.6 Conclusion	44
2 Apport des systèmes multi-agents	47
2.1 Introduction	47
2.2 Agent et systèmes multi-agents	47
2.2.1 Définition et propriétés d'un agent	47

2.2.2	Systèmes Multi-Agents	49
2.2.3	Organisation Muti-Agents (OMA) et principes d'interactions	51
2.2.4	Communication dans les systèmes multi-agents	54
2.2.5	De la typologie des agents aux applications des SMA	55
2.3	Apport des SMA dans l'ingénierie des SI	57
2.3.1	Agents d'Information pour un environnement d'information distribué	58
2.3.2	Agents Assistants pour une information personnalisée	63
2.3.3	Organisation types des Systèmes d'information orientés agents	64
2.3.4	Applications des SMA dans les SI dédiés au transport	71
2.4	Nouvelle OMA d'aide au déplacement multimodal	76
2.4.1	Méthodologies de conception des systèmes multi-agents	78
2.4.2	Du contexte transport vers une OMA d'aide au déplacement multimodal	80
2.5	Conclusion	88
3	Algorithmes de recherche et de composition	91
3.1	Introduction	91
3.2	Algorithmes de plus court chemin dans le domaine du transport	92
3.2.1	Algorithmes du plus court chemin classiques	92
3.2.2	Discussion	94
3.3	Algorithme de plus courts chemin distribué	95
3.3.1	Modélisation du problème	95
3.3.2	Principe du plus court chemin distribué	97
3.4	Algorithme de plus court chemin distribué et horaire	106
3.4.1	Modélisation du problème	106
3.4.2	Principe du plus court chemin distribué et Horaire	116
3.5	Algorithme de sélection du domaine de recherche	122
3.6	Conclusion	126
4	Implémentation et scénarios de simulation	129
4.1	Introduction	129
4.2	Implémentation et Simulation des algorithmes du SICM	130
4.2.1	Sénarios de simulation	130

<i>TABLE DES MATIÈRES</i>	<i>7</i>
4.2.2 Discussion	138
4.3 Plate-forme multi-agents pour le SICM	139
4.3.1 Vers une organisation holonique	139
4.3.2 Scénarios de déploiement	141
4.3.3 Implémentation	144
4.4 Conclusion	148
Conclusion et perspectives	151
Glossaire des abréviations	155
Bibliographie	156
Annexes	171

Table des figures

1	Evolution des modes de transport de personnes	16
1.1	Exemple de réseau multimodal multi-opérateurs	24
1.2	Exemple de réseaux régionaux et urbains [Atlas04]	25
1.3	Liaisons point à point entre différentes applications de l'entreprise	30
1.4	Communication entre les applications de l'entreprise via l'EAI	30
1.5	Urbanisation et EAI (selon France Telecom)	31
1.6	Architecture des Services Web	34
1.7	Stratégies d'intégration de l'information voyageur	35
1.8	architecture du système DELFI	39
1.9	Architecture du système Transport Direct	40
1.10	Principe de combinaison	42
1.11	Modèle conceptuel des données de la base de données chouette [Liu03b]	43
2.1	Représentation d'un système multi-agents selon Ferber	50
2.2	Formes d'interactions entre agents	51
2.3	OMA à structure hiérarchique	53
2.4	OMA à structure de marché	54
2.5	OMA à structure de communauté	54
2.6	OMA à structure de société	54
2.7	Principe d'un acte de communication	55
2.8	Collecte d'informations avec un agent mobile	61
2.9	Fonctionnement d'un agent interface assistant	64
2.10	Architecture médiateur-Adaptateurs [Shi02]	65
2.11	Architecture médiateurs-Adaptateurs du TSMMS [Shi02]	66

2.12	Architecture Interfaces-médiateur-Adaptateurs	68
2.13	Architecture Interfaces-Médiateurs-Adaptateurs	69
2.14	Architecture Interfaces-Coordinateur-Agents Mobiles-Adaptateurs	70
2.15	Organisation FIPA pour un système multi-agents d'information voyageur	73
2.16	Organisation SMA du système SCATEAgent [Yin et al.05]	74
2.17	Organisation MAPIS [Roze et al.04]	76
2.18	Agents d'Information	82
2.19	Agents du SICM	83
2.20	Comportement de l'Agent d'Information (AI)	84
2.21	Comportement d'Agent Courtier (AC)	85
2.22	Comportement de l'Agent Annuaire Selectionneur (AS)	86
2.23	Comportement de l'Agent Interface Utilisateur(AIU)	87
2.24	Comportement global du SICM	88
3.1	Algorithme Dijkstra de plus court chemin	93
3.2	Exemple d'un graphe $G(N, E)$ dans un environnement distribué	97
3.3	Exemple d'un graphe $G_{cint}(N_{cint}, E_{cint})$ dans un environnement distribué	100
3.4	Exemple d'un graphe d'intersection étendu $G_{eint}(N_{eint}, E_{eint})$	102
3.5	Algorithme de plus court chemin distribué	103
3.6	Exemple d'un modèle dépendant du temps	108
3.7	Exemple d'un modèle espace-temps	109
3.8	Propriété des réseaux FIFO	111
3.9	Algorithme de plus court chemin dans les réseaux dynamiques FIFO	112
3.10	Algorithme de plus court chemin dans les réseaux dynamiques non FIFO	114
3.11	Exemple d'un graphe partiel virtuel	117
3.12	Algorithme de plus court chemin distribué-FIFO	120
3.13	Algorithme de plus court chemin distribué et Horaire	121
3.14	Exemple d'un graphe d'adjacence d'AI	124
3.15	Algorithme d'inondation modifié	125
4.1	Exécution de l'algorithme d'inondation avec $k_T = 3$	131
4.2	Exécution de l'algorithme d'inondation	132
4.3	Ensemble des réseaux des SIAD associés au domaine de recherche	132

4.4	Itinéraire global relatif au domaine de recherche $AI_1 - AI_3 - AI_5$	133
4.5	Itinéraire global relatif au domaine de recherche $AI_1 - AI_2 - AI_4 - AI_5$	133
4.6	Domaine de recherche et graphe d'adjacence	134
4.7	Exemple d'un graphe virtuel partiel	135
4.8	Exemple d'un réseau de transport avec 4 lignes de bus	136
4.9	Organisation multi-agents du SICM	140
4.10	Organisation holonique	141
4.11	SICM avec un traitement centralisé	142
4.12	SICM personnalisés avec un traitement distribué	143
4.13	Réponse XML à une requête $req(A1, A6, 08 : 25)$	145
4.14	Réponse XML détaillée à une requête $req(A1, A6, 08 : 25)$	146
4.15	Description XML d'un service web	147
4.16	Exemple de communication des agents du SICM sur Jade	149

Liste des tableaux

2.1	Typologie des agents	57
3.1	Construction du G_{cint}	99
4.1	Horaires de passages du bus A, de A1 vers A9	137
4.2	Horaires de passages du bus A, de A9 vers A1	137
4.3	Chemin le plus rapide inter-opérateurs	138
..		

Introduction générale

Nous, nous déplaçons pour travailler, étudier, faire des courses, être à l'heure à tel rendez-vous, et quand on a le temps, nous nous déplaçons aussi pour nous faire plaisir, pour sortir, partir, et découvrir...

Pour répondre à ce besoin de mobilité, en agglomération ou ailleurs, on a recours à plusieurs modes de transport : voiture privée, transport en commun, vélo ou marche à pieds. Notre choix du mode de déplacement est souvent contraint par l'accessibilité de ce dernier, son prix, sa qualité, et la régularité du service proposé. Or, à ce jour la voiture privée reste, et continue à être le moyen de transport qui satisfait le plus, les différents critères des usagers. Il s'agit en effet d'un moyen de transport personnel qui ne nécessite pas un effort particulier de réflexion pour la préparation du voyage. Ce qui n'est malheureusement pas le cas des autres modes et notamment des transports publics.

Les statistiques de l'INSEE ¹(voir Fig 1) confirme en effet ce constat : le besoin de mobilité augmente et la voiture reste le moyen le plus utilisé dans les agglomérations.

Dans le cadre de la mobilité durable, l'heure est aujourd'hui à un usage moindre de la seule automobile au profit d'autres modes collectifs et notamment des transports en commun. Il devient alors nécessaire d'améliorer l'attractivité de ces modes collectifs pour promouvoir une utilisation rationnelle et pertinente du transport.

Or, pour améliorer cette attractivité, l'information voyageur, dite aussi multimodale, doit être au coeur de la réflexion et de l'action. En effet, un voyageur informé sur les modes disponibles, sur les horaires de départs et d'arrivées et sur les meilleurs itinéraires à prendre pour réaliser son déplacement, accède plus facilement aux services des transports publics.

¹www.ensee.fr

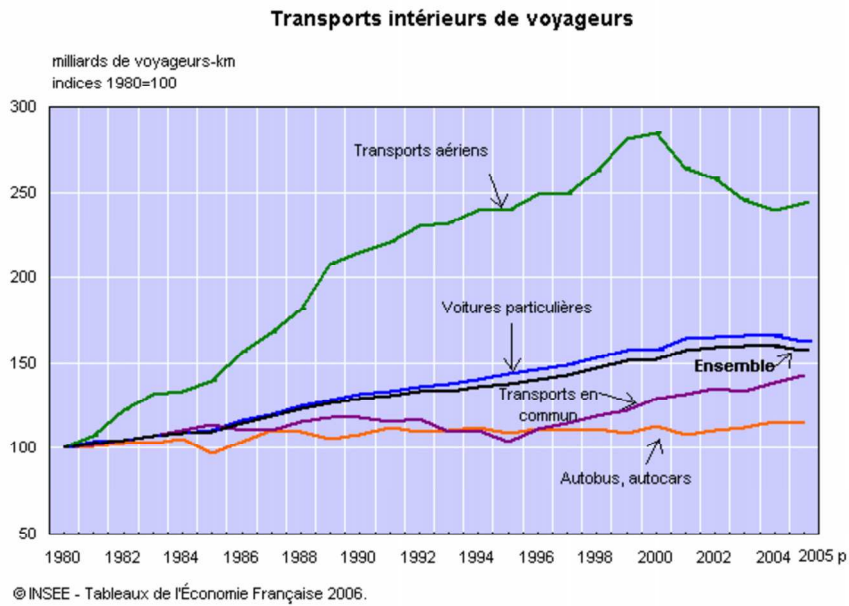


FIG. 1 – Evolution des modes de transport de personnes

Dans bien des cas, l'information multimodale, qui met en avant l'offre de mobilité sur un territoire donné, fait défaut. Ou alors, elle est éparpillée sur les sites web de différents opérateurs. Il conviendrait alors d'assurer la continuité du service de l'aide au déplacement sur un territoire donnée, en intégrant cette information multimodale et multi-opérateurs.

Afin d'éviter au voyageur de consulter plusieurs sites web d'opérateurs de transport en commun pour planifier son déplacement, ce travail vise à concevoir un système d'Information Coopératif de Mobilité (SICM) pour l'aide au déplacement multimodal. Il s'agit d'automatiser cette démarche de recherche et de composition d'itinéraires, pour fournir une information multimodale via un système intégrateur, en s'appuyant sur la théorie des systèmes multi agents (SMA) pour l'intégration et la médiation des systèmes d'information des différents opérateurs de transport.

Pour produire l'information multimodale nécessaire à l'aide au déplacement, le SICM doit accéder aux différents systèmes d'information des opérateurs de transport et intégrer des résultats de recherche qui sont générés par les différents calculateurs d'itinéraires des différents exploitants. Dans cette approche, le SICM est un intergiciel (middleware) qui devient un client parmi d'autres usagers des systèmes d'information existants.

Le SICM devient alors l'intermédiaire entre les différentes sources d'informations hétérogènes et distribuées d'une part et les clients d'autre part. Ce système doit être capable à la fois de trouver la bonne source d'information pour l'interroger selon les différentes requêtes des utilisateurs, et de regrouper les informations de manière cohérente pour répondre aux requêtes.

Afin de réaliser ce moteur de recherche et de composition en ligne d'itinéraires multimodaux, qui fournit un itinéraire composé mais surtout optimisé selon les critères de l'utilisateur, le recours à des algorithmes de plus courts chemins distribués " en-ligne ", et adaptés à des graphes dynamiques (dépendant du temps) a été retenu.

Le présent document est organisé de la manière suivante :

- Dans un premier chapitre, nous étudions le contexte global de l'information voyageur. Nous exposons aussi les architectures des systèmes d'information multimodale, et les techniques adoptées pour intégrer une information multi-opérateurs.
- Dans le deuxième chapitre nous étudions l'apport des systèmes multi-agents dans la conception et la réalisation des systèmes d'information. Nous proposons ensuite une organisation multi-agents dédiée à l'intégration de l'information multimodale, afin de réaliser le système d'information coopératif de mobilité (SICM).
- Les algorithmes de recherche et de composition d'itinéraires sont détaillée au troisième chapitre.
- Enfin, nous détaillons dans le dernier chapitre les applications et les scénarios de déploiement du SICM.

Chapitre 1

Systemes d'Information Multimodale : Du contexte actuel vers un cahier des charges plus adapté aux besoins des voyageurs

1.1 Introduction

L'information multimodale ou l'information voyageur, c'est l'information qui permet dans le transport public d'accompagner le voyageur avant et au cours de son déplacement. Cette aide au déplacement, est généralement composée d'une description détaillée d'offres d'itinéraires, des horaires de départ et d'arrivée, de durées de trajets, de coûts de tickets et ce concernant différents modes de transport et parfois concernant même différents opérateurs de transport. A cette information qui décrit essentiellement les itinéraires, s'ajoute une information dite annexe ou d'agrément, et qui permet selon les lieux de départ et d'arrivée de renseigner les utilisateurs, sur la météo, les restaurants à proximité des arrêts, les hôtels, ou sur d'autres services de proximité. Cette information annexe, n'est pas nécessaire pour décrire le déplacement en soi, mais elle peut être indispensable à l'utilisateur, pour mieux orienter son choix dans l'ensemble de l'offre de mobilité.

Par ailleurs, la qualité de l'information multimodale en termes de disponibilité et de pertinence devient aujourd'hui, un critère essentiel pour l'évaluation de la qualité du service

des opérateurs de transport. Mais même si, en France, la Loi d'Orientation sur les Transports Intérieurs (LOTI) existe depuis 1982 pour affirmer le droit à l'information voyageur, le développement actuel des applications dédiées reste proportionnellement limité au développement rapide des réseaux de transport et aux besoins croissant des usagers à ce type de service.

Aujourd'hui, chercheurs, industriels, et collectivités publiques s'impliquent dans un processus d'innovation, pour créer des services d'information multimodale et annexe et les adapter aux besoins des utilisateurs, et ce en considérant notamment le contexte industriel et commercial des différents opérateurs de transport.

Pour réaliser un état de l'art sur l'information multimodale, nous commençons dans ce chapitre par définir le contexte industriel du transport multimodal et multi-opérateurs. Cet aperçu nous permet de décrire ensuite une cartographie réelle des systèmes d'information pour le transport. A partir de cette cartographie, nous analysons les difficultés à surmonter pour pouvoir intégrer cette information à partir des systèmes d'information des différents opérateurs de transport. Nous nous intéressons ensuite aux approches techniques développées par les opérateurs, les régions et les pays pour offrir une information multimodale et multi-opérateurs.

1.2 Définition de l'information multimodale

Définissons tout d'abord les notions de multimodalité et d'intermodalité dans les transports [Liu03a] [Liu03b] :

- L'intermodalité est l'enchaînement de plusieurs moyens de transport pour un seul déplacement donné entre une origine et une destination.
- La multimodalité c'est l'offre de plusieurs moyens de transports pour un déplacement entre une origine et une destination.

L'information multimodale est l'information voyageur qui décrit des itinéraires intermodaux et multimodaux.

[Uster01] définit cette information comme étant celle qui propose un ou des itinéraires à l'utilisateur des transports, en tenant compte de l'ensemble de l'offre disponible pour une destination donnée ; et ce dans une optique de préparation de son déplacement, et d'offrir une

information pertinente et immédiatement opérationnelle, réduisant son incertitude pendant le voyage. L'information multimodale peut aussi décrire des déplacements sur un réseau de transport d'un opérateur ou sur les réseaux de plusieurs opérateurs de transport. Nous appelons alors un itinéraire intra opérateur, un itinéraire multimodal ou intermodal qui est assuré par un seul opérateur. Un itinéraire multi opérateur est par analogie l'enchaînement de plusieurs itinéraires intra opérateur. En définitif l'information multimodale est une information définie par son contenu : la description détaillée d'un itinéraire. Cet itinéraire peut être :

- intermodal ou multimodal
- intra-opérateur ou multi opérateur

Parallèlement, cette information peut être aussi définie par rapport à ses objectifs.

[Gude catalan et al.00] lui définissent 3 objectifs principaux :

- Faciliter le choix modal de l'utilisateur dans la panoplie des offres disponibles, et assurer la continuité des services.
- Faciliter l'usage des réseaux de transport en commun pour améliorer leur attractivité.
- Orienter si possible le comportement du voyageur vers une utilisation optimisée des infrastructures pour diminuer les congestions.

Enfin cette information multimodale, définie précédemment comme une description d'une offre d'itinéraire global sur un territoire donné composé de différentes offres locales, dépend étroitement du contexte du transport multimodal et multi opérateur, des systèmes d'information de transport adoptés par ces modes et ces opérateurs, et surtout des stratégies et des politiques de gestion de l'information, adoptées par ces derniers sur le dit territoire.

1.3 Gestion des réseaux de transport

1.3.1 Contexte et organisation des transports publics en France

Administrativement l'état Français est constitué de régions. Ces régions sont constituées de départements, qui sont eux même composés de communes. La réglementation des activités de transport, l'élaboration des schémas directeurs, l'initiation des projets dans ce domaine et le contrôle de leurs applications sont assurés par ces composantes représentatives de l'état (régions, départements, commune) dites Autorités Organisatrices (AO).

L'état est alors considéré comme l'Autorité Organisatrice des services de transports d'intérêt national.

La région constitue l'Autorité Organisatrice des services de transport ferroviaires et routiers d'intérêt régionaux. Elle initie, finance, suit et contrôle les projets de transport régionaux. Et, il est de même pour les départements et communes, qui gèrent respectivement les projets départementaux de transport public dits "non urbain", et communaux dits "urbain".

Par ailleurs, il est aussi, à préciser qu'il n'y a aucune hiérarchie entre ces AO. A titre d'exemple, pour créer une nouvelle ligne de train départementale, le département n'a pas à impliquer la région à laquelle il appartient. Il est d'ailleurs, le seul initiateur et financeur du projet. Pour l'exploitation de cette ligne l'AO a le choix entre une exploitation directe dite régie et la délégation de services publics. Dans ce dernier cas, le département lance un appel d'offre, auquel vont répondre, différentes sociétés exploitantes de transport (SNCF, Transpole) sur la base d'un cahier des charges prédéfini au départ.

Les AO restent propriétaires des équipements et des infrastructures de transport (bus, parcs de dépôts, rames, lignes de métro). Les exploitants gèrent l'exploitation du service.

La majorité des transports publics urbains sont exploités par des sociétés privées, ce qui donne lieu à des conventions de délégation des services public, qui précisent les engagements et les responsabilités réciproques des deux parties.

Dans le cas où l'AO gère directement son réseau de transport, l'information voyageur qui n'est autre qu'une description de l'offre du service du transport est considérée comme propriété de l'AO. Par ailleurs, si c'est un exploitant privé qui gère totalement ou partiellement ce réseau de transport-ce qui est généralement le cas-, alors ce dernier est propriétaire de l'information sur le service qu'il gère, et les bases de données de cet exploitant sont protégées par le droit d'auteur.

En résumé, les AO initient et contrôlent les projets de transport. Ils sont propriétaires des réseaux de transport mais ce sont les exploitants privées qui ont, en général, le monopole de l'information voyageur. De ce fait créer une information voyageur multimodale et multi-opérateurs, revient inévitablement à créer et intégrer une information multi-exploitants. C'est la problématique de l'information multimodale en France : à ce jour, ce sont les AO qui poussent ces propriétaires à coopérer pour créer une information multimodale.

1.3.2 Contexte et organisation des transports publics en Europe

En Europe, c'est aussi le cadre institutionnel et juridique et surtout la manière d'exploiter les réseaux de transport qui détermine le degré de développement de l'information multimodale. Ainsi au Royaume Uni, la première autorité organisatrice est le marché de mobilité [Savy et al.01]. En effet, même si l'état impose une coordination entre les différentes collectivités en termes de politique de transport, et même si la hiérarchie est également imposée entre ces différentes collectivités, ce sont les opérateurs privés qui cherchent à s'organiser selon les périmètres qu'il trouvent plus profitables, où le marché est plus intéressant. L'état intervient alors de manière indirecte dans l'organisation du transport : il peut financer des lignes de bus ou de métro qui sont jugées déficitaires ou allouer des aides tarifaires pour une classe d'utilisateur (seniors, étudiants...). Dans ce cas l'information multimodale et multi-opérateurs est forcément une information multi-exploitants. Mais à la différence du cas Français, c'est le marché qui pousse les opérateurs à créer ce type de service. Nous montrons plus tard dans cette étude, que ceci aide d'une certaine manière au développement des services de l'information multimodale.

1.3.3 Configuration d'un réseau multimodal multi opérateur

Dans ce paragraphe, nous allons définir le domaine de l'information multimodale à partir d'une représentation de la configuration d'un réseau multimodal multi-opérateurs.

Le réseau de transport sur une zone géographique étendue, n'est autre que la juxtaposition de plusieurs sous-réseaux. A titre d'exemple, le réseau de transport public de France est une juxtaposition de plusieurs réseaux de transport régionaux, et d'un réseau national (SNCF) qui permet de lier entre ces réseaux via des stations communes dites pôles d'échange. Un réseau régional peut aussi être défini comme la juxtaposition de différents réseaux départementaux qui sont liés par un ensemble de lignes de bus (autocar), et de TER régionaux. C'est aussi le cas des communes. Dans les agglomérations urbaines, la gestion du réseau est généralement associée à un opérateur qui exploite plusieurs modes de transport sur cette zone. La continuité du service de transport est assurée entre ces modes via des stations d'intermodalité dites pôles d'échange [James et al.02].

Un pôle d'échange est dit intermodal s'il permet le transfert des voyageurs d'un mode de transport vers un autre, il est dit de surcroît inter-opérateurs s'il permet le transfert des

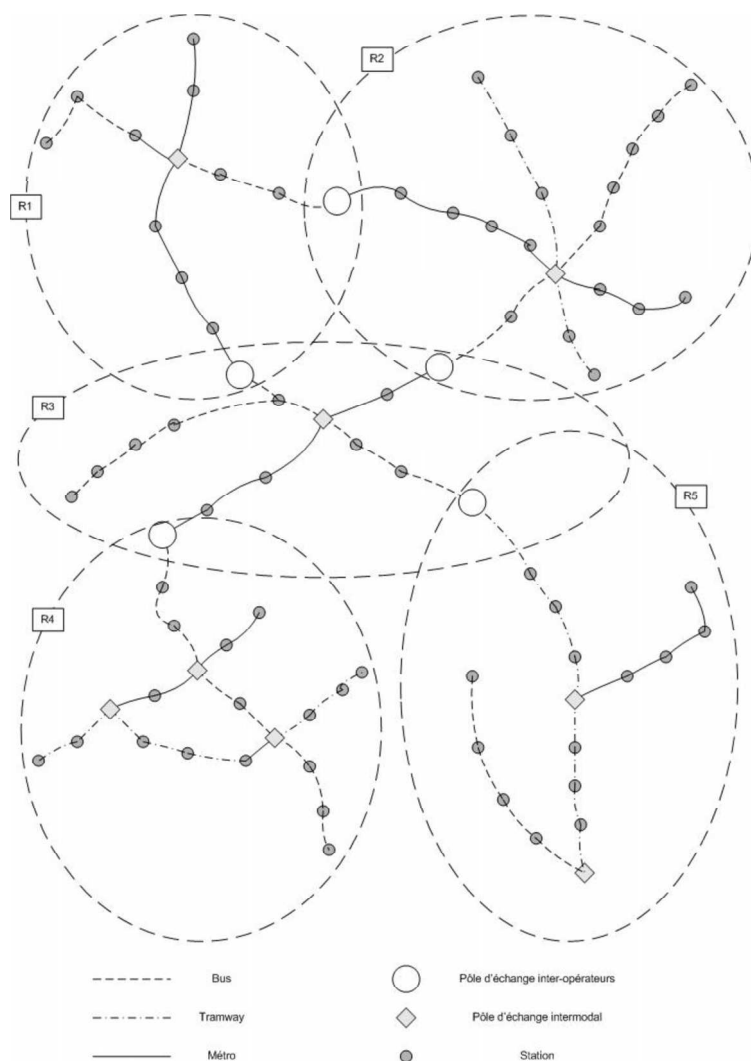


FIG. 1.1 – Exemple de réseau multimodal multi-opérateurs

voyageurs du réseau de transport d'un opérateur vers le réseau d'un autre. La figure 1.1 illustre le réseau multimodal de transport public de la région Nord Pas de Calais. L'agglomération Lilloise et l'agglomération Valenciennoise disposent chacune de leur propre réseau multimodal de transport public (Bus, Tramway et métro). Ces deux agglomérations sont liées au réseau régional TER de la SNCF via des pôles d'échanges (gares SNCF), qui sont en même temps des arrêts de bus, de tramway ou de métros pour les réseaux locaux.

Chaque exploitant dispose de l'information multimodale concernant le réseau de transport qu'il exploite. Ainsi, pour produire une information multimodale sur une zone géographique



Réseau de transport de la région Nord Pas De Calais



Réseau urbain de L'agglomération Lilloise



Réseau urbain de l'agglomération valenciennoise

FIG. 1.2 – Exemple de réseaux régionaux et urbains [Atlas04]

étendue, il est nécessaire d'impliquer les différents opérateurs dont les réseaux de transport couvrent, au moins en partie, la zone considérée.

1.4 Systèmes d'Information pour le transport

Cette partie est dédiée à la présentation de l'état de l'art sur les systèmes d'information multimodale. Il s'agit de décrire l'existant et d'énumérer les difficultés de production pour aboutir dans la suite de l'étude à des éventuels scénarios de production et d'amélioration de cette information.

1.4.1 Définition d'un Système d'information

Un système d'information est selon l'ISO, [Afnor97] un "système de traitement de l'information accompagné des ressources organisationnelles associées, telles que les ressources humaines, techniques et financières, et qui fournit et répartit des informations" pour les éventuels utilisateurs.

la plupart des opérateurs de transport disposent, en effet, d'un SAE, un Système d'information d'Aide à l'Exploitation, qui leur permet de gérer toute leur activité de transport [Wang06]. Ces systèmes offrent un ensemble de fonctionnalités, et notamment l'aide à l'information des voyageurs :

- une aide à la régulation des services même en cas de perturbations.
- une aide à la sécurité des employés et des clients.
- une aide à la billettique.
- une aide à la maintenance.
- une aide à l'information des voyageurs.

De ce fait l'information voyageur a été toujours, produite et diffusée par un SAE. Le Système d'Information d'Aide au Déplacement (SIAD), n'est autre, dans la majorité des cas, qu'un module de diffusion d'un SAE.

Par ailleurs, un système d'information multimodale est un système d'information d'aide au déplacement (SIAD) qui implique plusieurs opérateurs de transport, pour fournir aux usagers une information multimodale qui renseigne sur différents modes de déplacements possibles dans un périmètre géographique donné[Perreau02].

1.4.2 Cartographie de l'information multimodale à l'image de la configuration des réseaux

La majorité des systèmes d'information d'aide au déplacement (SIAD) pour le transport en commun en France sont des systèmes monomodaux, qui offrent une information sur le seul moyen de transport qu'ils exploitent. Nous citons à titre d'exemple le système d'information de l'opérateur Still des bus de Lens¹et celui de la compagnie Filbleu pour le réseau de bus de Tours².

¹<http://www.still.fr>

²<http://www.filbleu.fr>

D'autres systèmes d'information sont multimodaux, et offrent une information portant sur plusieurs modes de transport, mais généralement cette information concerne un seul opérateur, comme par exemple le site Transpole, opérateur de la métropole lilloise³, celui de l'opérateur Transvilles de l'agglomération Valenciennoise⁴, le site web RATP⁵ ou celui de l'AMIVIF⁶ pour la région de l'Île-de-France, Le Pilote pour l'agglomération marseillaise⁷, ou la SNCF pour le réseau national de France⁸. Parce qu'ils appartiennent généralement à un seul opérateur de transport, ces systèmes multimodaux restent limités géographiquement [Roze03].

D'autres systèmes d'information intègrent, en plus des modes courants des transport public (train, tramway, metro, bus, ...), d'autres modes tels que les taxis pour les transports de Douai⁹, ou le vélo pour l'exploitant de l'agglomération Cleremontoise¹⁰.

A l'image de la configuration des réseaux de transport, la cartographie de l'information d'aide au déplacement multimodal, n'est autre qu'une juxtaposition de systèmes d'information existant. A la différence, des réseaux de transport eux mêmes, dans le cas des systèmes de transport, cette juxtaposition entre les différents systèmes d'information n'implique pas un échange d'information multimodale ou une coopération entre eux.

1.4.3 SIM : spécification des besoins

Un citoyen souhaite avoir à disposition une information sur l'offre de mobilité, sur les itinéraires respectant ses critères. La recherche de cette information peut se faire par le client lui même sur les différents SIADs existants. Mais, cela nécessite une expertise forte en termes de connaissance de l'offre générale de transport, des différents sites d'information des opérateurs (Transpole, SNCF, RATP), et de la tarification applicable par chaque exploitant. Cela nécessite, aussi, une certaine disponibilité de la part du client et même une certaine maîtrise de l'outil informatique. Pour lui éviter cette tâche fastidieuse de recherche, de mémorisation

³<http://www.transpole.fr>

⁴<http://www.transvilles.com>

⁵<http://www.ratp.fr>

⁶<http://www.transport-idf.com>

⁷<http://www.lepilote.fr>

⁸<http://www.sncf.fr>

⁹<http://www.transportsdudouais.fr>

¹⁰<http://www.www.moovicite.com>

des différentes éventualités, de combinaison de différents moyens de transport et pour l'aider dans cette démarche de prise de décision, industriels et chercheurs travaillent aujourd'hui sur l'offre d'une information multimodale. Il s'agit de fournir toute information utile et pertinente sur les différents modes de déplacement (métro, tramway, bus, voiture, etc.), afin d'une part d'améliorer le confort et l'efficacité des trajets à un niveau individuel, et d'autre part pour favoriser l'usage multimodal et raisonné des différents modes de transport à l'échelle collective

L'idée générale de ce travail de thèse est d'élargir le champ de cette information multimodale et de "la faire coïncider avec un territoire défini au delà de certaines limites territoriales et des champs de compétences des exploitants" [Gude catalan et al.00]. Ceci permettra au voyageur, qui ignore totalement les cloisonnements institutionnels des réseaux et technologiques des systèmes d'informations, d'avoir une visibilité globale sur l'ensemble de l'offre de mobilité pour se déplacer librement. L'objectif est d'avoir un gisement de données ou un système d'information multimodale global intégrant et actualisant l'ensemble de l'offre des transports collectifs ou même individuels sur un territoire élargi.

1.4.4 Freins à la production de l'information multimodale

Les freins de production de l'information multimodale sont selon [Danflous02] et [Jamed et al.02] :

- Organisationnels : Différents opérateurs sont impliqués dans ce processus de production, ce qui nécessite une coopération efficace et des efforts non négligeables d'organisation et de coordination.
- Economiques : La production de cette information est coûteuse, ce qui pousse les opérateurs à chercher des modèles économiques pour son exploitation en tant que service. Par ailleurs dans ce contexte global de compétitivité, cette information présente aussi une valeur commerciale puisqu'elle permet d'attirer les usagers sur le réseau de transport de l'opérateur qui la diffuse.
- Juridiques : La mise à disposition et l'utilisation de cette information requièrent une réflexion réelle non seulement sur la propriété de cette information mais aussi sur la responsabilité liée à son utilisation.
- Techniques : Les sources d'informations sont distribuées sur différents serveurs. Les applications et les bases de données sont hétérogènes. Les technologies utilisées évoluent

continuellement.

- Qualité : La mise à jour de cette information, la prise en compte des éventuelles perturbation du réseau, l'optimisation des temps de réponses aux requêtes des utilisateurs, fournir une information concise précise et personnalisée constituent les défis majeurs pour produire une information multimodale de qualité.

Face à ces difficultés, deux alternatives s'imposent aux chercheurs :

- soit nous attendons la disparition de ces difficultés, et on entame alors une recherche basée sur des hypothèses simplificatrices telles que : la disponibilité de cette information voyageur ou l'établissement dans un futur proche de ce gisement de données dit global.
- soit nous proposons en plus des algorithmes de calcul d'itinéraires, des architectures techniques et des scénarios d'intégration qui prennent compte de toutes les difficultés citées précédemment.

En effet nous essayerons de répondre à travers ce travail de recherche à la question suivante : " Comment intégrer l'information voyageur de chaque opérateur pour offrir une information multimodale qui décrit un itinéraire global?".

1.5 Intégration des systèmes d'information

Avant d'étudier l'état de l'art sur l'intégration de l'information voyageur, il s'est avéré nécessaire de faire une veille technologique sur les techniques, les standards, et les protocoles utilisés actuellement dans l'industrie dans le domaine de l'intégration de l'information.

1.5.1 Veille technologique sur l'intégration de l'information : EAI

1.5.1.1 EAI (Intégration des Applications d'Entreprise) et Urbanisation des systèmes d'information

Intégrer l'information hétérogène dispersée devient aujourd'hui un enjeu majeur pour diverses entreprises. En effet, pour mener efficacement son processus métier, une entreprise a non seulement besoin de communiquer avec ses partenaires (fournisseurs et clients), mais avec ses propres filiales ou entités : à savoir échanger les informations entre les différents services et départements (Achat, Direction, R&D,..).

Par ailleurs, depuis le milieu des années 80, les systèmes d'information des entreprises

étaient conçus sur la base des SGBDs (système de gestion de bases de données) propriétaires et hétérogènes. Pour que tous les systèmes d'information existant à l'intérieur ou à l'extérieur de l'entreprise, puissent communiquer des données mais aussi des traitements, et pour continuer à exploiter ces systèmes existants, les industriels procèdent aujourd'hui à une intégration d'applications, appelée couramment EAI (Entreprise Application Integration). 30 % des dépenses informatique actuelles sont affectées à ce procédé [Chevauss04].

L'intégration de l'information devient aujourd'hui une intégration de systèmes d'information qui passe inévitablement par une intégration d'applications (EAI) via la conception de middlewares dits aussi intergiciels. Ces intergiciels vont assurer la liaison entre les différentes applications de manière à éviter une explosion de liaisons et de communications point à point (Fig 1.3 et Fig 1.4).

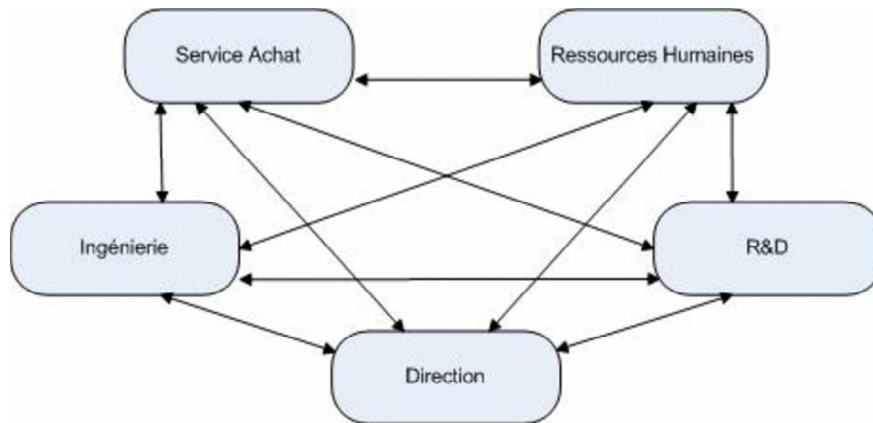


FIG. 1.3 – Liaisons point à point entre différentes applications de l'entreprise

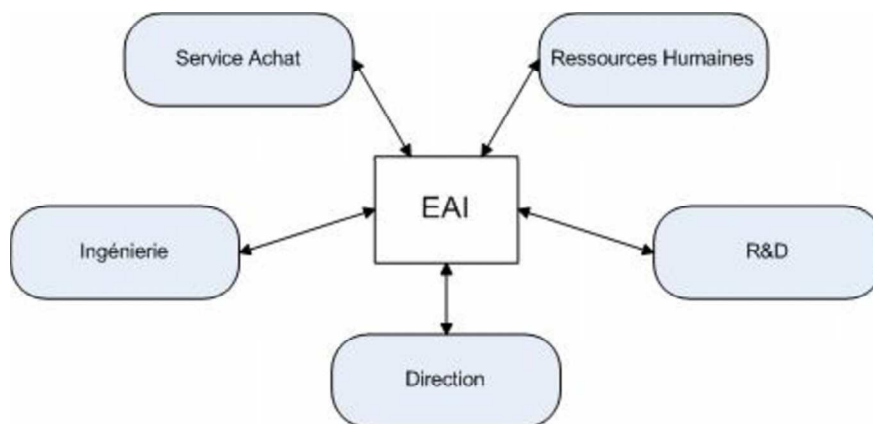


FIG. 1.4 – Communication entre les applications de l'entreprise via l'EAI

Actuellement, dans le contexte industriel actuel, l'entreprise développe au cours de son activité, et pour son métier, des applications locales et diverses dont les interdépendances en termes de données et de traitements, restent souvent indéfinies. Avant de procéder à la phase d'intégration de ces applications, une phase d'analyse est nécessaire pour déceler les interactions possibles entre les différentes composantes fonctionnelles. Il s'agit d'analyser pour proposer une architecture globale du système d'information qui le rend plus modulaire et par conséquent plus évolutif et plus adapté à une EAI.

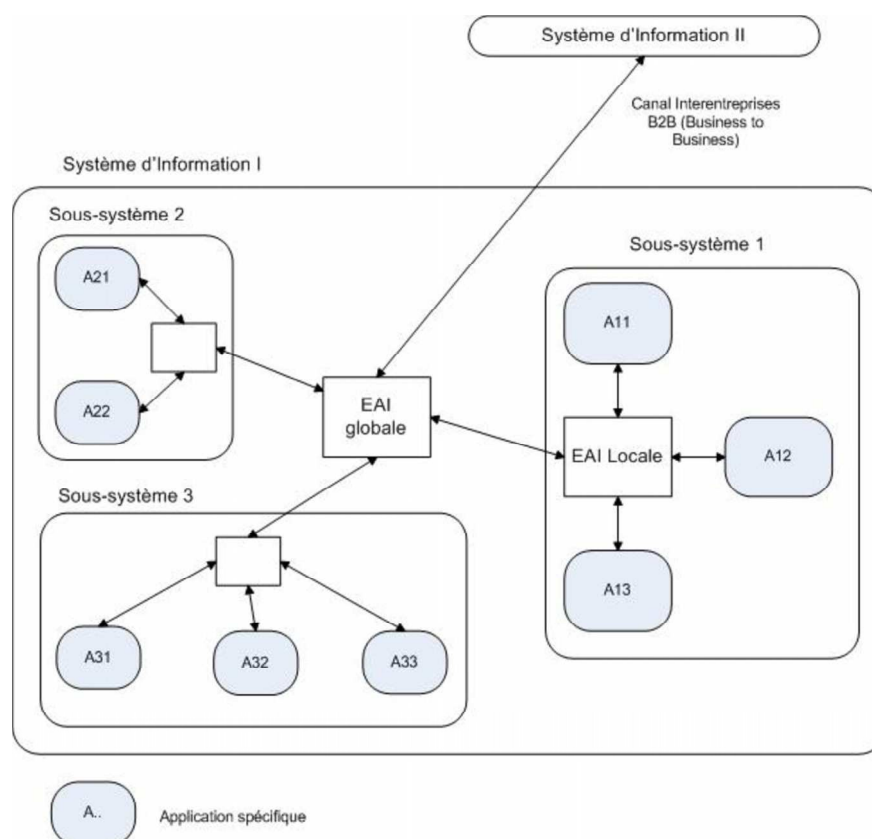


FIG. 1.5 – Urbanisation et EAI (selon France Telecom)

Ce processus est dit "urbanisation d'un système d'information", il est défini par [Chevauss04] comme suit : "Comme dans une ville, l'urbanisme d'un système d'information décrit des quartiers, des zones et des liaisons. Il en résulte une cartographie recensant toutes les applications, les bases de données et leurs interdépendances".

Dans un système global l'urbanisation va permettre d'isoler les sous-systèmes avec leurs liaisons, et interactions. L'EAI va construire l'infrastructure qui va permettre de les relier et de les présenter comme un système global.

1.5.1.2 Modes d'intégration de l'information dans l'EAI

Nous distinguons aujourd'hui différents types d'intégration

- Une intégration au niveau utilisateur : Il s'agit généralement des portails web où l'intégration se fait au niveau de l'interface utilisateur, qui va assurer l'identification ("logging") des utilisateurs à différentes applications en même temps.
- Une intégration au niveau application : Il s'agit d'une infrastructure qui s'appuie sur la technique des messages pour recevoir et envoyer des messages de données d'une application vers une autre, avec des fonctions de transformation et de routage de données. Cette gamme de middlewares supporte une panoplie d'API (Application Programatic Interface) qui permettent d'accéder, via des bibliothèques prédéfinies, aux différentes applications à intégrer.
- Une intégration de données : Il s'agit d'intégrer et de fédérer les bases de données,
 - soit pour réaliser une base de données centrale, par l'importation des données des autres bases, et une adaptation de leurs différents modèles de données,
 - soit pour réaliser une base de données distribuée.

1.5.1.3 XML et EAI

XML (eXtensible Markup Language) est par définition un langage extensible de description de données. Créé en 1996 ¹¹, il permet une description générale des données à base de balises. A la différence du HTML, XML sépare les balises de contenu et de mise en forme.

Il est principalement un métalangage de description de structure et de contenu qui permet aujourd'hui d'assurer le transport et l'échange de flux de données entre différentes applications d'un système d'information via une description préalable et formalisée du contenu.

Les principaux apports de l'XML à l'EAI [Chevauss04][Chauvet03] :

- il permet la portabilité des données et leur partage entre applications très diverses, du fait qu'il est auto-descriptif : son contenu décrit aussi la structure des données qu'il transporte.
- normalisé depuis 1998 par le consortium mondial de l'internet (World Wide Web Consortium) qui regroupe plusieurs industriels et chercheurs, il devient aujourd'hui un outil standard pour les différents projets d'intégration d'applications.

¹¹<http://www.w3.org/xml/>

- Séparant le contenu de la mise en forme, il facilite la transformation des données.

1.5.1.4 Services Web et EAI

Un service web est un composant logiciel métier, effectuant un certain traitement, et accessible à distance via le protocole HTTP (Internet). Il est le successeur de la technologie CORBA (Common Object Requester Broker Architecture) de l'OMG, de la technologie COM (Commun object model) de Microsoft, et de la technologie EJB (Entreprise Java Beans) de Sun, et qui permettent tous d'invoquer une méthode ou un traitement à distance, via le réseau, dans une architecture orientée objet et distribuée. Par rapport aux technologies précédentes qui ne permettent pas de construire une architecture répartie hétérogène composée d'objets de différents technologies (objets COM, CORBA...)[Melliti04], les services web permettent d'intégrer des composants applicatifs hétérogènes en standardisant les interfaces (API) et les protocoles de communication via Internet[Fleury05]. En effet, ils sont construits sur la base de trois standards technologiques :

- SOAP (Simple Object Access Protocol) : C'est un standard du W3C, qui définit un protocole d'échange de documents XML via Internet. C'est le langage de communication entre une application cliente qui invoque le service et une application fournisseur qui répond à des requêtes et offre le service. Il s'agit d'une enveloppe XML ¹² qui va contenir et transmettre la requête du client ou la réponse du fournisseur.
- WSDL (Web Service Description Language) : Il s'agit aussi d'un standard du W3C, défini en XML, et qui permet de décrire de manière détaillée et standardisée l'ensemble des services publiés par le service web : l'ensemble des méthodes à invoquer à distance avec leurs paramètres d'entrée et leurs valeurs de retour.
- UDDI (Universal Description Discovery and Integration) : Norme du comité OASIS¹³, elle définit la structure de l'annuaire répertoriant différents services web publiés par des entreprises. L'annuaire UDDI est composé de pages blanches répertoriant les entreprises fournisseurs de services, de pages jaunes décrivant les services web publiés, et de pages vertes décrivant comment invoquer ces services web via le WSDL.

L'architecture basée sur les services web dite aussi SOA (Service Oriented Architecture) est défini par trois entités avec différents rôles (Fig 1.6) :

¹²www.w3.org

¹³www.oasis-open.org

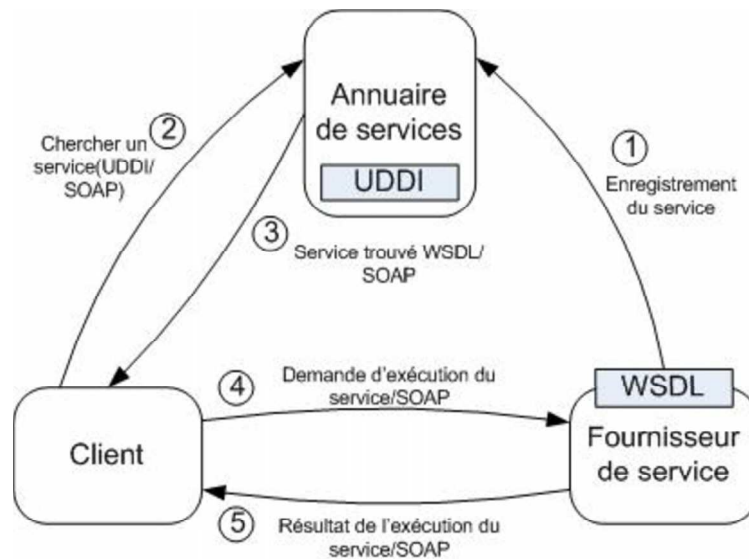


FIG. 1.6 – Architecture des Services Web

- Un fournisseur de service : il s'agit de l'entreprise, ou de l'acteur qui va publier un service web métier et l'inscrire dans l'annuaire (UDDI).
- un annuaire : qui sert de pages jaunes, pour les clients cherchant un service web.
- Un client qui va chercher un web service, -correspondant à ses besoins- sur l'annuaire, pour ensuite l'invoquer, et l'intégrer dans son propre processus métier.

Les services web constituent aujourd'hui l'infrastructure la plus utilisée dans le domaine de l'intégration de l'information, puisqu'elle permet de construire une architecture ouverte, qui permet d'intégrer facilement de nouveaux composants.

1.5.2 Intégration de l'information voyageur : Etat de l'art

1.5.2.1 Stratégies d'intégration

Le Système d'Information d'Aide au Déplacement (SIAD), dont disposent la majorité des opérateurs de transport est généralement composé d'une Base de Données (BD) et d'un Algorithme de Calcul d'Itinéraires (ACI) qui constitue le traitement pour la recherche du meilleur chemin. La base de données décrit via une information statique, telle que l'ensemble des lignes, des arrêts, des itinéraires et des horaires de passage, l'ensemble du réseau et services offerts à l'utilisateur. L'ACI utilise cette base de données locale pour rechercher et trouver le ou les itinéraires qui s'adaptent au mieux à la requête de l'utilisateur. Pour

produire l'information multimodale, et afin de la composer à partir de plusieurs SIAD de différents opérateurs, il est nécessaire de procéder à une intégration d'information. Deux stratégies d'intégration sont alors à considérer (Fig 1.7) :

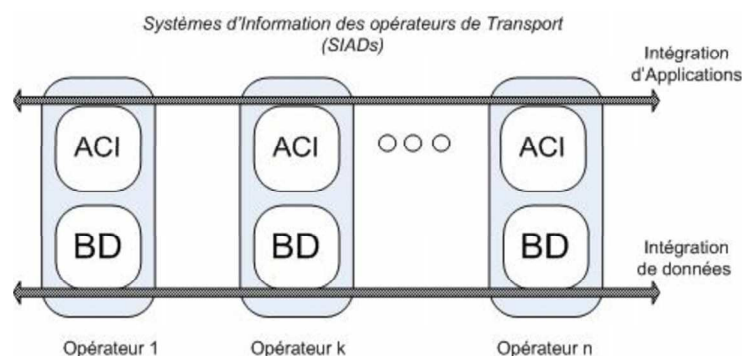


FIG. 1.7 – Stratégies d'intégration de l'information voyageur

- Soit, on procède à une intégration des différentes bases de données, pour constituer une base de données globale et centralisée : un gisement de données conforme à un modèle relationnel unique ; et construire, par la suite, un ACI global et assez puissant pour s'adapter à cette base de donnée centralisée et globale. Il s'agit là d'une intégration de données.
- Soit, on procède à une intégration d'applications : produire et composer l'information multimodale à partir des résultats de plusieurs ACI. En d'autres termes, il s'agit d'interroger les SIAD existants d'une manière automatique, et de profiter des algorithmes et des calculateurs d'itinéraires qui sont déjà implémentés, pour composer une offre d'itinéraire global qui répond à la requête de l'utilisateur.

1.5.2.2 Les données nécessaires pour un service d'information multimodale

la création d'un service d'information multimodale par une intégration des systèmes d'information des différents opérateurs existants, nécessite un échange de données entre les exploitants à impliquer, et ce pour les deux types d'intégration cités précédemment : pour une base de données centralisée ou pour une intégration d'applications. En effet, pour construire un service qui offre la description d'un itinéraire global multimodal et multi-opérateurs, il faudrait disposer de la description des offres des différents opérateurs (arrêts et horaires de passages,...). Par ailleurs, pour récupérer ces offres de tronçons d'itinéraires, il faudrait

aussi disposer des données nécessaires pour formuler les requêtes intermédiaires (horaires de départs et d'arrivées dans les pôles d'échanges...) Les données nécessaires à ce service sont définies comme suit [Wang06] [Jamed et al.02].

Données statiques Il s'agit des données qui permettent de définir le réseau de transport d'un opérateur (le référentiel Transport) :

- les données qui permettent de définir les arrêts, avec leurs géolocalisations.
- les données qui permettent de décrire l'offre du service de transport proposé, à savoir les modes utilisés, les itinéraires, ainsi que les horaires théoriques de passage des véhicules du réseau par les différentes stations (tables de marche).
- les données relatives aux arrêts intermodaux d'un même opérateur et celles relatives aux pôles d'échange inter-opérateurs et qui constituent les interfaces avec les autres opérateurs.
- les données relatives au positionnement des arrêts par rapport à des localisants qui permettent d'interfacer le transport en commun avec d'autres modes (marche à pied , voiture privée (VP)). Ceci dépend de l'intégration ou non d'un service de déplacement porte à porte (adresse à adresse) ou d'un service qui permet d'intégrer le mode VP et les parkings.

Données dynamiques Il s'agit principalement de données temps réel qui permettent de mettre à jour l'information sur l'offre du service disponible, telles que les modifications des durées de trajets théoriques ou l'annonce de perturbations ou d'interruptions de services. Ces données dites dynamiques sont plus difficiles à inclure dans une base de données centralisée puisqu'elles nécessitent une mise à jour permanente.

1.5.2.3 Normalisation de l'information de transport

Pour procéder à un échange de données il est nécessaire de définir non seulement les données à échanger mais aussi leur modèle conceptuel et le format d'échange à adopter par les différents intervenants. En effet, les travaux actuels s'intéressent principalement à la normalisation de ces deux aspects. Nous citons dans la suite des normes et des projets de normes Européennes qui la standardisation des modèles et des formats de données.

TRANSMODEL La norme TRANSMODEL ¹⁴ initiée en France en 2001, est devenue aujourd'hui une norme Européenne. A sa version 5.1, elle définit un modèle conceptuel de données qui permet de décrire le référentiel "transport public". Destinée aux opérateurs du transport public, cette norme leur permet de gérer toutes les données relatives à leurs activités de planification, de régulation, et d'exploitation[Liu03a] :

- Conception des horaires
- Gestion du personnel roulant
- Suivi et contrôle en temps réel
- Information des usagers
- Perception des titres de transport
- Tableaux de bord / statistiques

La norme est définie par un ensemble de diagrammes UML et Merise détaillés, qui sont généralement utilisés lors de la conception de logiciels ou de bases de données des SAE. La norme est utilisée actuellement par des opérateurs européens en France, en Belgique, en Italie, au Royaume Uni, aux Pays bas, et par quelques autorités organisatrices du Danemark.

TRIDENT TRIDENT¹⁵ (Transport Intermodality Data Sharing and Exchange Networks)

La commission de normalisation CN03 pour les applications télématiques dans les transports publics, miroir du groupe européen CEN TC278 / WG3, a créé en 2002 un groupe de travail pour normaliser les échanges de données concernant l'information voyageur entre exploitants et Autorités Organisatrices de transport[Liu03b]. Le groupe a constaté que le projet Européen TRIDENT, coordonné par ERTICO¹⁶ (Réseau Européen pour les systèmes de transport intelligents), constitue la meilleure base pour l'élaboration d'une normalisation. En effet TRIDENT, décrit un ensemble de schémas XML, qui permettent de normaliser l'échange de données entre exploitants du VP mais aussi du TC, et ce en mode normal, mais aussi en mode perturbé. Pour le mode VP, TRIDENT se base sur DATEX (format d'échange pour l'information routière). D'autant plus que pour le transport public, les schémas XML se basent sur un modèle objet en UML qui décrit le référentiel transport : Il s'agit de TRANSMODEL qui a déjà été adopté comme norme au niveau Européen.

¹⁴<http://www.transmodel.org/>

¹⁵<http://www.ertico.com/en/activities/activities>

¹⁶<http://www.ertico.com>

Les spécifications de TRIDENT sont utilisées en France par la RATP pour la gestion des situations perturbées et pour l'échange de données, essentiellement d'horaires théoriques, avec l'AMIVIF¹⁷ (Association Multimodale d'Information des voyageurs en Ile de France).

SIRI SIRI (Service Interface for Real Time Information)¹⁸, est un projet Européen qui a été initié en 2004, par l'Allemagne, la France et le Royaume Uni, et le Comité Européen de normalisation pour normaliser l'échange de données en temps réel entre différents serveurs d'opérateurs de transport. L'information échangée concerne les tables de marches, les véhicules et les horaires de départs possibles dans les pôles d'échange. SIRI est un protocole d'échange de données de transport avec des requêtes/réponses en XML. Il se base aussi sur TRANSMODEL comme modèle conceptuel de données.

1.5.2.4 Projets Européen et Internationaux

Plusieurs systèmes ont été développés en Europe pour intégrer et fournir l'information multimodale. Les deux exemples les plus cités dans la littérature sont l'exemple Allemand avec leur système DELFI et l'exemple du Royaume Uni avec le système Journey web. Il s'agit dans les deux cas d'une architecture répartie via une intégration d'application. D'autre pays, de moins de surface, comme les PaysBas ou la Suisse ont opté plutôt pour une architecture centralisée via une intégration de données.

DELFI : L'exemple Allemand Initié en 1996 en Allemagne, le projet DELFI¹⁹ dont le nom signifie "tables de marche horaires électroniques et continues", vise à intégrer les différents calculateurs d'itinéraires locaux des différents opérateurs relatifs aux divers états ("landers") de la République Fédérale. Il s'agit en effet d'une intégration d'application via des interfaces ou API standardisées utilisant le protocole CORBA [Danflous00]. Pour répondre à la requête d'itinéraire d'un utilisateur, DELFI commence par déterminer les deux calculateurs locaux relatifs à l'origine et à la destination du trajet. Ensuite, le système interroge les serveurs de ces deux opérateurs ainsi que le serveur de l'opérateur dit "National" ou "de longue distance" pour récupérer les pôles d'échange intermédiaires. Un composeur principal

¹⁷<http://www.transport-idf.com>

¹⁸<http://www.kizoom.com/standards/siri/index.htm>

¹⁹<http://www.delfi.de>

va concaténer les offres recueillies, pour composer et construire différentes alternatives d'un itinéraire global. Ces offres seront par la suite présentées à l'utilisateur. L'évolution actuelle de ce projet vise l'intégration de TRIDENT comme norme d'échange de données.

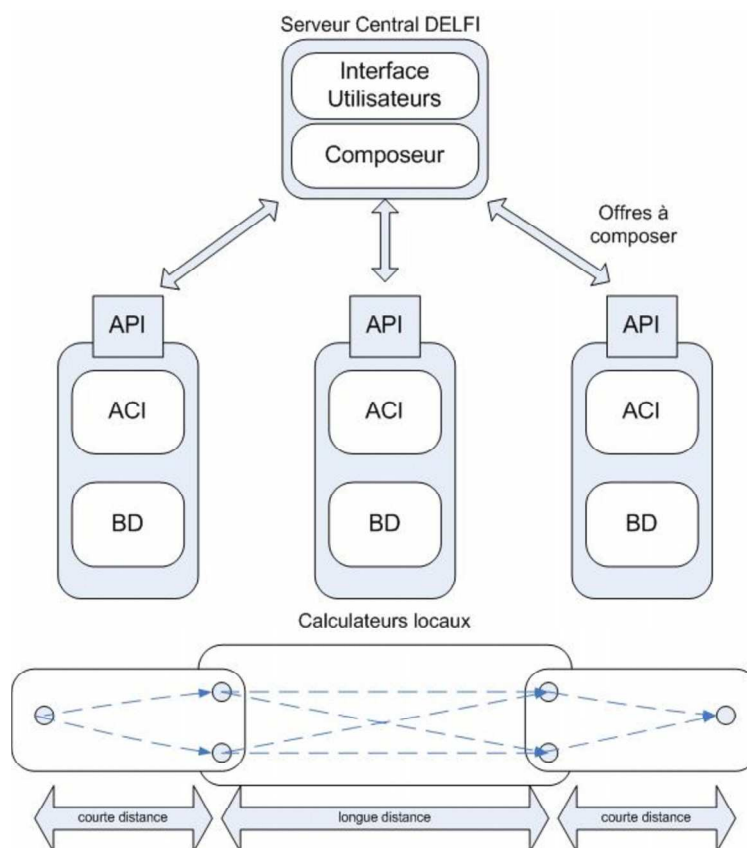


FIG. 1.8 – architecture du système DELFI

Netzwerk Direct²⁰ est un autre projet Allemand qui reprend l'architecture de DELFI et remplace l'infrastructure d'intégration CORBA, par une autre basée sur XML/SOAP.

JourneyWeb : L'exemple Anglais Le projet Transport Direct²¹ a été initié en 2000 par le ministère de transport public Anglais pour réaliser un système national d'information multimodale qui gère non seulement l'information routière mais aussi l'information relative au transport en commun. Dans le cadre de ce projet, un protocole de communication JourneyWeb²² a été mis en place pour permettre aux systèmes d'information des différents

²⁰<http://www.netzwerk-direct.de>

²¹<http://www.transportdirect.info>

²²<http://www.journeyweb.org.uk/>

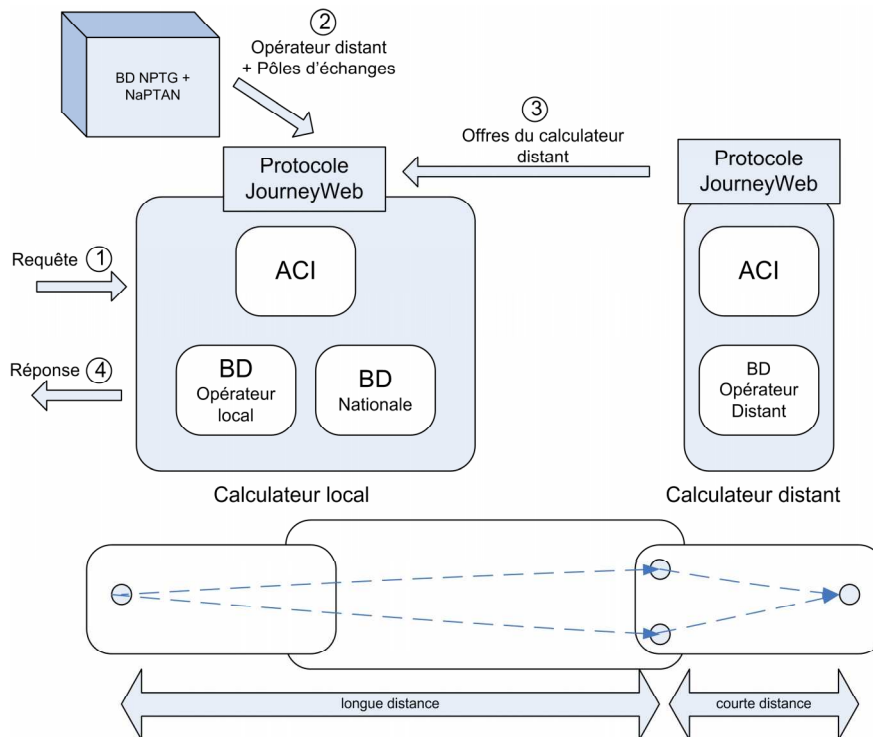


FIG. 1.9 – Architecture du système Transport Direct

opérateurs du transport public, de s'interroger mutuellement, afin que chacun puisse offrir à ses clients un itinéraire global multimodal et multi-opérateurs. Les données échangées sont en format XML et conforme à un standard local Transexchange. Pour son fonctionnement JourneyWeb utilise deux bases de données :

- NPTG (National Public Transport Gazetteer) : Il s'agit d'une base de données nationale répertoriant régions, villes, villages, et surtout des pôles d'échange nationaux qui permettent de basculer d'un réseau régional vers le réseau National.
- NaPTAN (National Public Transport Access Node) : C'est aussi une base de données nationale, répertoriant l'ensemble des noeuds (arrêts et stations) qui permettent l'accès à un réseau de transport d'un opérateur local et ce pour les différentes régions.

Dans le projet Transport Direct chaque calculateur local est capable de rechercher un itinéraire d'un point de départ inclus dans sa zone de couverture vers un pôle d'échange national qui permet l'accès au réseau de l'opérateur distant relatif à la destination choisie. Ceci suppose que chaque calculateur local dispose de toute l'information concernant le réseau national. Ainsi, pour répondre à la requête d'un utilisateur, un calculateur local va consulter

le NPTG et le NaPTAN pour déterminer le calculateur local distant correspondant à la destination et les pôles d'échanges qui permettent d'y accéder. Ensuite il va composer un itinéraire global en concaténant son offre à celle du calculateur de l'opérateur distant, voir figure 1.9.

GOFAS : l'exemple Suisse A l'opposé des deux exemples précédents, la Suisse dont le territoire est moins étendu, a opté pour une intégration de données. Le système d'information multimodale national réalisé, appelé GOFAS, est un système centralisé [Fan et al.99]. Il est composé de deux sous-systèmes :

- Infopool : il permet de réaliser l'importation des données horaires et géographiques à partir des bases de données des différents opérateurs pour construire une base de données centralisée qui couvre tout le territoire national.
- Internet-GIS : c'est le calculateur d'itinéraires, qui utilise la base de donnée centralisée.

9292 : l'exemple des Pays-Bas Le système 9292²³ est un portail national hollandais pour l'aide au déplacement multimodal. Il informe aussi sur les perturbations possibles du service. C'est un système centralisé, qui procède à l'importation des données des différents opérateurs par des convertisseurs spécifiques.

EU-Spirit Le projet Européen EU-Spirit ²⁴ initié par ERTICO, reprend l'architecture répartie de DELFI pour fournir une information voyageur multimodale et porte à porte, et ce en utilisant les transports collectifs et ferroviaires. Le système couvre les réseaux de transports de trois pays : l'Allemagne, le Danemark et la Suède. Par ailleurs, à la différence de Delfi, les pôles d'échange qui permettent d'accéder aux réseaux des différents exploitants dans ces trois pays, ainsi que les itinéraires de longues distances (TGV, Thalys...) qui permettent de les lier, sont stockés antérieurement dans une base de données centrale du serveur actif qui assure la composition. De ce fait, pour composer un itinéraire global ou "international", suite à la requête d'un utilisateur, le système commence par déterminer les calculateurs régionaux ou nationaux qui correspondent au lieu de départ et au lieu d'arrivée demandés, ensuite il interroge ces serveurs sur les pôles d'échanges qui permettent d'accéder à ces stations.

²³www.9292ov.nl

²⁴www.eu-spirit.com

Une fois les pôles d'échange déterminés, le système propose des itinéraires "longue distance" internationaux-à partir de sa base centrale- et qui permettent de lier ces pôles. Ensuite il va interroger les calculateurs relatifs à l'origine et à la destination du trajet, pour compléter cet itinéraire international par des itinéraires nationaux ou régionaux qui permettent de lier les pôles d'échange à l'origine et à la destination (voir Fig 1.10).

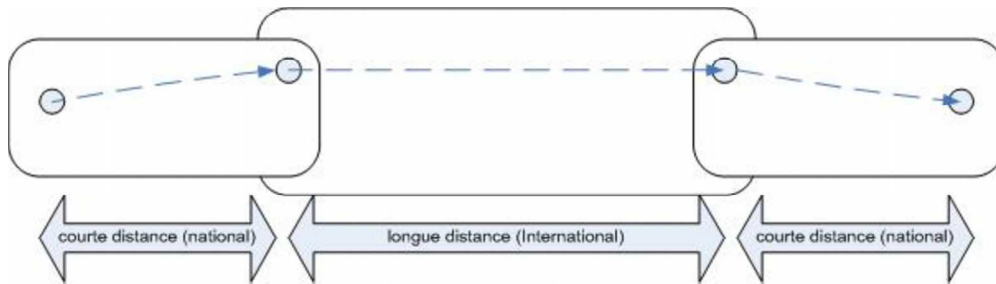


FIG. 1.10 – Principe de combinaison

1.5.2.5 Projets Nationaux

Pour répondre à ce défi de développement de l'Information Voyageur en France, fédérer les projets et capitaliser les recherches et les expériences en termes d'information multimodale, le PREDIT²⁵, Programme de Recherche et de l'Innovation dans les Transports Terrestres a créé la PREDIM²⁶, Plate-forme de Recherche et d'Expérimentation pour le Développement de l'Information Multimodale.

La plate-forme PREDIM constitue non seulement un dispositif de mutualisation d'expertises mais aussi un espace de collaboration pour le montage de nouveaux projets entre exploitants et collectivités. Ces projets de recherche visent essentiellement à :

- Améliorer la complémentarité des différents modes de transport et de déplacement, tant individuels que collectifs, en proposant de nouveaux concepts, services, prototypes et systèmes pour le développement de l'information multimodale.
- Créer des services concrètement applicables et des prototypes diffusables industriellement, et qui prennent en considération le contexte et l'organisation du transport en France.

²⁵<http://www.predit.prd.fr>

²⁶<http://www.predim.org>

La PREDIM a ainsi initié, plusieurs projets régionaux qui visent à produire l'information multimodale via l'intégration de l'information provenant de différents opérateurs de transports d'une même région [Gendre99]. Nous assistons ainsi à l'émergence de plusieurs SIM régionaux, qui reposent généralement sur le modèle d'intégration de données dans une base de données intermédiaire et centralisée.

Pour faciliter ce type d'intégration, la PREDIM a aussi accompagné la création par le CERTU²⁷ de l'outil CHOUETTE [Liu03a][Liu03b]. Il s'agit d'une API, un adaptateur, qui permet de convertir [Isorce et al.05a] différents formats de données pour ensuite les intégrer dans une base de données centrale conforme aux spécifications TRANSMODEL et TRIDENT. Cette base de données est utilisée ultérieurement par un éventuel calculateur d'itinéraires. Cependant l'outil n'intègre qu'une partie des spécifications TRIDENT relatives à la description des horaires et des informations statiques décrites par le diagramme UML illustré par la figure 1.11. Cet outil ne gère pas les informations dynamiques.

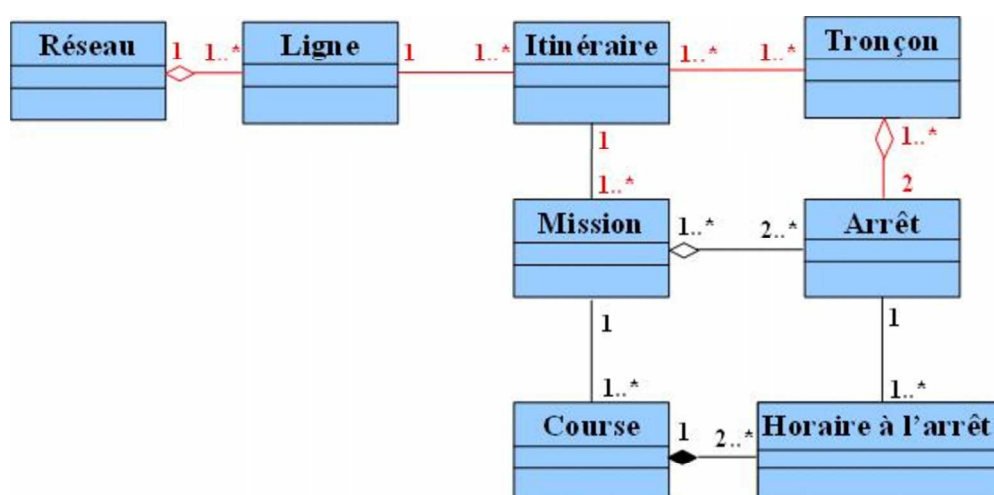


FIG. 1.11 – Modèle conceptuel des données de la base de données chouette [Liu03b]

La PREDIM a aussi initié un projet PASSIM²⁸ qui vise à réaliser un Prototype d'Annuaire des Sources et Service d'Information Multimodale, pour répertorier les sites d'information des différents opérateurs de transport en France et ce par département et par région. Cependant, il n'existe pas à ce jour une politique générale et nationale en France pour l'intégration de l'information multimodale. Certes nous assistons à un développement continu des SIMs

²⁷Centre d'Études sur les Réseaux de Transport et l'Urbanisme

²⁸<http://passim.predim.org>

régionaux mais ceci ne va pas dans le sens d'une création d'un SIM national global pour les transports collectifs.

Par ailleurs la PREDIM a aussi accompagné un projet MTSYSTEM pour la "standardisation d'un calcul d'itinéraires global" [Isorce et al.05a][Isorce et al.05b][Isorce et al.05c]. Il s'agit d'une reprise du concept du système Anglais Journey Web, mais avec des échanges de données qui se font via les services web. Une approche intéressante dans la mesure où [Isorce et al.05a] essayent d'adapter le standard technologique des services web aux standards du métier de transport : TRIDENT et TRANSMODEL et ce en s'inspirant du modèle de données utilisé par chouette. Le système vise aussi à composer un itinéraire global à partir des différentes offres recueillies des différents calculateurs locaux.

1.6 Conclusion

En conclusion, la problématique de l'intégration de l'information multimodale reste à ce jour un sujet d'actualité puisque les solutions techniques proposées sont souvent adaptées à un contexte et à une certaine organisation des transports collectifs dans un territoire donné et surtout limité. La centralisation de l'information multimodale par la création d'un gisement de données partagé, est une solution qui a été adoptée, dans nombreux projets d'intégration, nous citons l'exemple de la Suisse des Pays-Bas et des SIMs régionaux en France. Cette solution, bien qu'elle soit réalisable et fonctionnelle, ne permet pas d'assurer la mise à jour de l'information, et l'intégration des nouveaux opérateurs d'une manière automatisée. En plus cette mise à jour nécessite des opérations fréquentes d'imports de données, et donc une maintenance permanente de la base de donnée centralisée.

La solution distribuée qui se base sur une intégration d'applications, et sur la composition en ligne d'un itinéraire global multimodal et multi-opérateurs par un intergiciel- dit aussi un compositeur d'itinéraire- reste à ce jour la solution qui s'adapte au mieux à l'organisation et au contexte industriel des transports collectifs. En effet cette solution permet d'exploiter les solutions existantes, de les faire coopérer, et de garantir une implication minimale, en termes de développement de la part des différents opérateurs. Cette approche s'appuie, par ailleurs, sur des projets de pré-normes pour la standardisation des échanges de données TRIDENT et SIRI. Elle permet aussi de construire un système d'information ouvert qui peut intégrer facilement les nouveaux opérateurs et qui laisse la responsabilité de la mise à jour de cette

information aux différents exploitants qui la diffusent. Mais bien qu'elle soit bien adaptée au contexte actuel des transports collectifs, sa mise en oeuvre reste un défi technologique et scientifique. En effet, même si les technologies d'aujourd'hui permettent de construire des applications interopérables et des middlewares pour l'intégration des applications, l'organisation d'un SIM distribué, et ouvert reste à définir, et les algorithmes de recherche et de composition d'itinéraires en ligne restent à développer. Nous nous proposons alors d'utiliser l'approche multi-agents pour définir et modéliser une architecture flexible, d'un SIM global distribué, complexe et multi-acteurs, en détaillant les scénarios de coopérations et de communications à utiliser, par les différents agents du système pour l'intégration de l'information voyageur. Ensuite, nous essayerons d'étudier les algorithmes à utiliser pour cette recherche et composition en ligne.

Chapitre 2

Apport des systèmes multi-agents : Organisation et agents d'information

2.1 Introduction

Dans le chapitre précédent nous avons présenté un état de l'art sur les SIM existants, et sur les méthodes d'intégration de l'information utilisées à ce jour pour créer et fournir l'information multimodale d'aide au déplacement multimodal et multi-opérateurs. Dans ce chapitre nous allons introduire les agents et les systèmes multi-agents, et préciser surtout leurs apports dans l'ingénierie des systèmes d'information. L'étude de différents cas de leur utilisation dans le domaine de l'intégration de l'information et notamment le domaine de l'information de transport, nous permettra par la suite de proposer des organisations possibles d'agents d'information qui permettent de rechercher et de composer l'information multimodale et multi-opérateurs à partir de différents SIAD.

2.2 Agent et systèmes multi-agents

2.2.1 Définition et propriétés d'un agent

Dans la littérature scientifique, plusieurs définitions existent pour définir un agent. Nous optons pour celle de Mandiau [Mandiau et al.02a], qui le définit comme une "entité qui agit dans un environnement". L'agent est capable de percevoir "au moins partiellement" son

environnement, et d'agir en fonction de cette perception. Ses actions sont alors le résultat d'un raisonnement rationnel et d'une planification préalable.

D'une manière plus détaillée, Ferber [Ferber95] définit un agent comme une entité physique ou virtuelle :

- qui est capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- qui est mue par un ensemble de tendance (sous la forme d'objectifs, individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- qui possède des ressources propres,
- qui est capable de percevoir (mais de manière limitée) son environnement, qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- qui possède des compétences et offre des services,
- qui peut éventuellement se reproduire,
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Par ailleurs un agent peut aussi être défini par rapport à ses propriétés qui sont essentiellement :

- l'autonomie est définie par Demazeau et Muller [Demazeau et al.90] comme une existence indépendante : un agent autonome est un agent dont l'existence ne dépend pas de celle des autres. Une autre définition de l'autonomie [Russel et al.95] stipule qu'un agent autonome est un agent qui agit sans l'intervention des humains ou des autres agents, et qui a de ce fait un contrôle sur ses actions et sur ses états internes. Un agent logiciel autonome est une entité logicielle qui, à la différence de l'objet, peut choisir de répondre ou de ne pas répondre à un éventuel appel à ses méthodes ou à ses compétences. L'agent est ainsi capable de prendre des initiatives sans avoir recours à des interventions externes.
- la réactivité d'un agent est sa capacité de percevoir et de réagir aux modifications survenues dans son environnement. Le choix des actions et de l'ordre dans le quel elles vont être entreprises, se fait en fonction de cette perception.
- la proactivité d'un agent, c'est sa capacité à se fixer ses propres buts et objectifs par

ses propres initiatives.

- la continuité d'un agent c'est sa capacité à être en permanence actif sans avoir à être provoqué par un stimulus externe. Florez définit cette continuité comme la persistance d'une identité et d'un état sur une longue période et ce à la différence des objets logiciels [Florez99]. En effet ces agents logiciels sont souvent implémentés sur des processus légers dit "threads", qui s'exécutent en parallèles.
- la sociabilité c'est la capacité d'un agent de communiquer et d'interagir avec d'autres agents, soit pour échanger des informations nécessaires à son fonctionnement interne, soit pour faire aboutir des mécanismes de coopérations et /ou de coordination mutuelles des différentes actions mutuelles.

2.2.2 Systèmes Multi-Agents

Un système multi-agents est défini par Mandiau [Danflous00] comme étant "un ensemble d'entités qui coordonnent leurs connaissances, buts, expériences et plans pour agir ou résoudre des problèmes, incluant le problème de coordination inter agents lui-même". Il est défini d'une façon plus détaillé par Ferber [Ferber95] comme étant un système composé de :

- un environnement,
- un ensemble d'objets passifs pouvant être perçus, créés, modifiés ou détruits par des agents,
- un ensemble d'agents actifs,
- un ensemble de relation qui relie les objets entre eux,
- un ensemble d'opérations ou de compétences offrant la possibilité aux agents de percevoir, produire, consommer, transformer et manipuler des lois de l'environnement,
- un ensemble de lois qui sont des opérateurs chargés de représenter l'application des action des agents sur le monde et la réaction de ce monde à ces actions et qu'on appellera des lois universelles.

La figure 2.1 illustre la définition de Ferber.

Une autre définition citée par Fayech [Fayech03], et qui s'inspire de celle de [Geen et al.97], énonce que :

Un système multi-agents est un réseau d'agents (solveurs) faiblement couplés qui coopèrent ensemble pour résoudre des problèmes qui dépassent les capacités ou les connais-

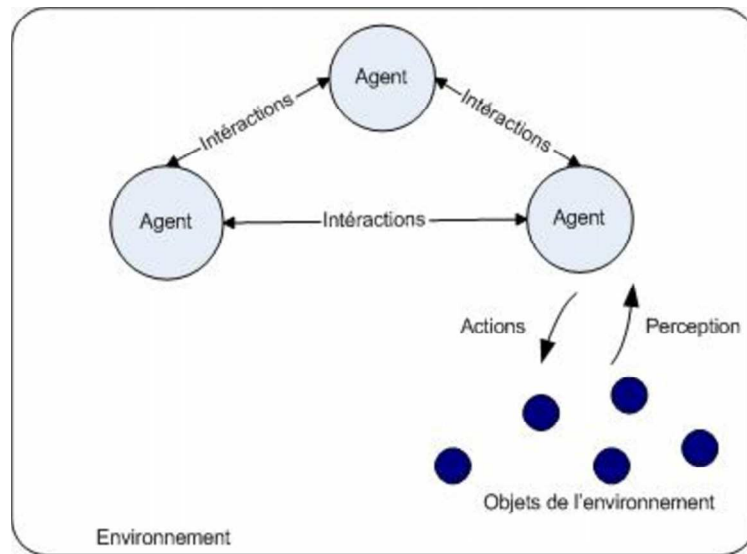


FIG. 2.1 – Représentation d'un système multi-agents selon Ferber

sances individuelles de chaque agent. Les agents sont autonomes et peuvent être de natures hétérogènes.

Ainsi, les agents d'un SMA, n'ayant pas une visibilité globale sur leur environnement, ils ne peuvent avoir qu'un champ d'actions limité sur l'ensemble des objets de cet environnement. De ce fait pour résoudre un problème global, ces agents sont amenés à coopérer et à communiquer pour échanger des informations et pour mieux coordonner leurs actions individuelles et locales.

La notion d'organisation des agents dans un SMA, s'impose pour définir les rôles des agents et les mécanismes de coordination et de communication entre eux.

Par ailleurs un SMA, en terme d'architecture peut être¹ :

- Ouvert : Les agents y entrent et en sortent librement. C'est l'exemple des clients qui entrent et sortent dans une place de marché.
- Fermé : Le nombre des agents reste toujours le même. C'est le cas d'un match de foot.
- Homogène : Tous les agents sont construits sur le même modèle, et implémentés de la même façon. C'est le cas des colonies de fourmis.
- Hétérogène : Les agents ne sont pas construits sur le même modèle et sont implémentés différemment.

¹notes de cours SMA de Joel Quinqueton, LIRMM, Montpellier, France

2.2.3 Organisation Muti-Agents (OMA) et principes d'interactions

L'organisation d'un système multi-agents, définit l'architecture globale du système. Elle précise aussi pour chaque agent son rôle et ses fonctions par rapport au groupe, et les règles d'interaction à adopter dans son environnement. Ces organisations d'agents logiciels s'inspirent des organisations des sociétés humaines et animales, et ce pour la définition des rôles et des interactions.

2.2.3.1 Principes d'interaction dans les organisations SMA

Pour résumer les différents types d'interactions possibles entre agents, [Weiss00] propose le schéma suivant.

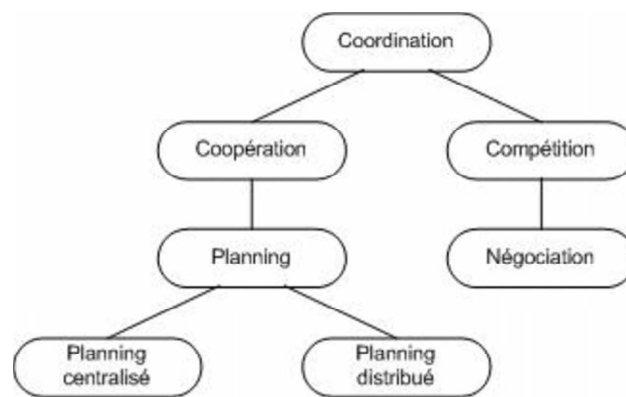


FIG. 2.2 – Formes d'interactions entre agents

[Weiss00] définit la coordination comme "la propriété d'un système composé d'au moins deux agents, exécutant des actions dans un environnement partagé". Cette notion de ressources partagées dans l'environnement implique la nécessité de la coordination. Les agents devraient coordonner leurs actions individuelles avec les autres pour aboutir à l'objectif global du groupe. Cette coordination permet alors :

- d'éviter les situations de conflits par la négociation pour les agents antagonistes (ayant des buts et des objectifs contradictoires),
- d'améliorer l'efficacité et l'utilité de chaque agent par la coopération pour les agents non antagonistes.

Négociation C'est une méthode de coordination qui permet à plusieurs agents, d'atteindre -suite à un processus de communication et d'échange d'informations- un accord mutuel pour

entreprendre une action donnée d'une certaine manière. Elle induit, par cette communication, des relaxations de buts initiaux, des concessions mutuelles, des mensonges ou des menaces [Fayeche03]. Elle est donc considérée comme une méthode de résolution de conflits et de recherche de consensus.

Coopération D'après [Weiss00], la coopération est la coordination parmi des agents non antagonistes, qui cherchent à se satisfaire mutuellement sans se gêner. Cette coopération, initiée par un échange d'information, est souvent associée à la notion de collaboration. la collaboration est une forme d'interaction qui étudie la manière de répartir le travail, et par conséquent l'allocation de tâches, entre plusieurs agents.

En effet l'allocation de tâches peut se faire dès la conception du système multi-agents, en définissant une organisation de résolution de problèmes qui est non adaptable. Mais dans d'autres cas, on peut avoir une allocation de tâches qui se fait d'une manière dynamique suite à un processus de coopération entre agents, de manière à ce que la planification des actions individuelles à entreprendre par chaque agent, se fasse aussi d'une manière dynamique.

2.2.3.2 Typologie des organisations dans les SMAs

Pour définir la typologie des organisations SMA dans la littérature, nous nous intéressons principalement aux travaux de [Mandiau et al.02a] et à ceux de [Adam00]. En effet, le premier classe ces organisations selon son processus de formation. Le deuxième les classe selon les niveaux hiérarchiques établis entre les différents agents.

Mandiau [Mandiau et al.02a] distingue deux modèles d'organisation fondamentalement différents :

1. L'organisation émergente : Il s'agit d'une organisation qui n'est pas prédéfinie dès la conception du système, mais qui se définit "graduellement" avec les interactions et les échanges inter-agents. Une structure organisationnelle apparaît ou émerge à partir des comportements individuels. Ce type d'organisation fait, généralement, intervenir un grand nombre d'agents et ne dispose pas d'un dispositif pour le contrôle global : "chaque entité, par son action locale, participe au développement de l'ensemble sans qu'aucune d'entre elle ne dirige les autres" [Mandiau et al.02a].
2. l'organisation support d'activités : c'est une organisation qui s'inspire des organisations

humaines dans les sociétés et notamment dans les entreprises. En effet la structure organisationnelle est prédéfinie dès la conception du système. Elle précise le rôle, les compétences, et les possibilités d'interactions associés à chaque agent. Elle impose aussi un contrôle global au système, qui peut être centralisé (un agent coordinateur) ou décentralisé sur différents agents.

Adam [Adam00] classe autrement les organisations multi-agent. S'appuyant sur les travaux de [Le strugeon95], il se base sur la typologie de leurs interactions. Il distingue :

- les OMA à structure hiérarchique, sont des structures rigides où le contrôle est centralisé sur un seul agent qui communique des ordres à des exécutants. l'environnement est schématisé par plusieurs niveaux hiérarchiques. Et un exécutant peut être un chef hiérarchique pour un niveau inférieur. Ce type d'OMA vise à satisfaire un objectif unique imposé par l'agent chef hiérarchique.
- les OMA à structure hétérarchique qui présentent aussi différents types :
 - les organisations de type marché présentent plusieurs coordinateurs possibles dans un même niveau, pour différents agents exécutants. Le contrôle est alors décentralisé sur les différents agents coordinateurs qui visent différents objectifs opérationnels : tâches à réaliser. C'est le cas de l'organisation de la plateforme MAGIQUE [Adam00]. Chaque agent coordinateur va lancer un appel d'offres vers les autres agents. Ensuite, il va récupérer les offres et établir un contrat de transaction avec le plus offrant.
 - les organisations de type communauté, où le contrôle est distribué sur les différents agents qui possèdent les même capacités et dont les interactions se résument à des échanges de flux de données.
 - les organisations de type société, sont le croisement entre l'organisation hiérarchique et l'organisation de communauté, avec des règles d'interaction à préciser.

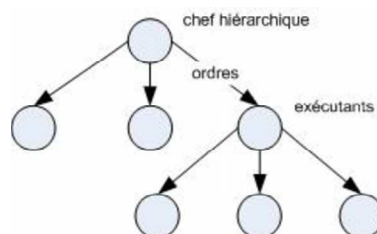


FIG. 2.3 – OMA à structure hiérarchique

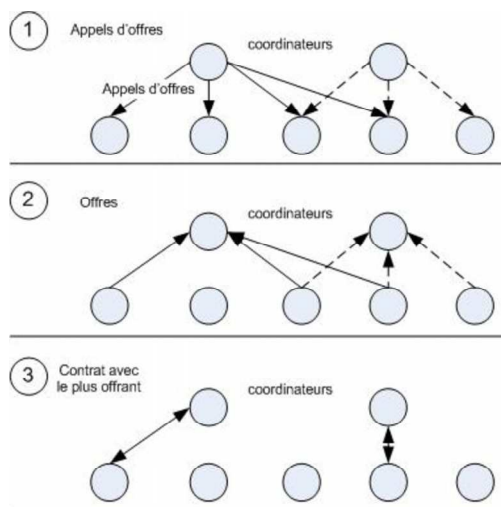


FIG. 2.4 – OMA à structure de marché

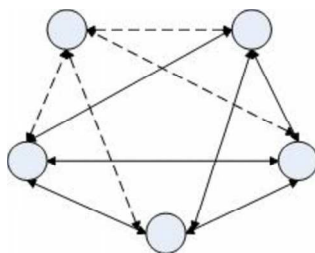


FIG. 2.5 – OMA à structure de communauté

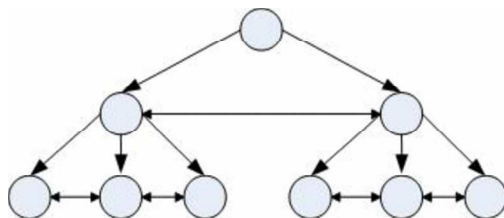


FIG. 2.6 – OMA à structure de société

2.2.4 Communication dans les systèmes multi-agents

Pour se coordonner entre eux, pour négocier, et pour coopérer les agents ont besoin de communiquer. Cette communication peut être directe ou indirecte :

- La communication indirecte se fait en apportant des modifications sur l’environnement. Ainsi un agent émetteur effectue des modifications sur l’environnement ; l’agent récepteur perçoit la modification et interprète le message. L’agent émetteur peut propager des signaux, ou laisser des traces dans l’environnement.
- La communication directe se fait via l’envoi de messages entre agents, elle est souvent associée à une action.

Cet acte de communication directe consiste en une transmission d’informations d’un émetteur vers un récepteur. Pour faciliter cette transmission, l’information est codée à l’aide d’un langage, par l’émetteur, et elle est décodée par le récepteur dès son arrivée [Badeig05]. Ceci est illustré par la figure suivante.

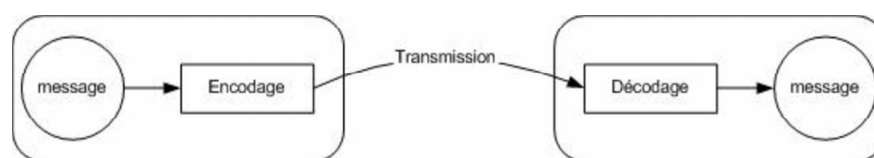


FIG. 2.7 – Principe d’un acte de communication

Pour la communication entre agents, deux langages sont couramment utilisés. KQML (Knowledge Query and Manipulation Language) développé à l’université du Maryland (USA) en 1993 et ACL (Agent Communication Language) développé en 1993 par la FIPA² (Foundation for Intelligent Physical Agents), d’après [Roze03].

2.2.5 De la typologie des agents aux applications des SMA

La typologie des agents a fait l’objet de plusieurs travaux de recherche [Bradshaw97], [Le strugeon95] et [Geen et al.97]. Tous ces travaux s’inspirent de la typologie présentée par [Nwana96]. En effet celle-ci classe les différentes sortes d’agents dans un espace multidimensionnel selon leurs différentes propriétés.

Ainsi les agents peuvent être réactifs ou cognitifs :

- Les agents réactifs sont des agents qui réagissent rapidement, et selon des règles prédéfinies, à des stimuli extérieurs. Ils sont constamment en état de veille sur les changements de leur environnement. Leurs actions rapides et non-réfléchies sont similaires

²www.fipa.org

à des réflexes. Ces agents sont souvent utilisés dans la **simulation de mondes et de société** : Il s'agit de simuler le comportement de sociétés existantes (humaine : match de football, animales : société d'abeilles, ou de machines : robots mobiles), et d'étudier l'impact du comportement et des propriétés individuelles des agents sur l'évolution globale du groupe.

- Les agents cognitifs sont des agents réfléchis. Leurs actions sont souvent le résultat d'une planification et d'un raisonnement rationnel qui se base sur un ensemble de connaissances sur leur environnement, sur les autres agents et sur eux mêmes [Mandiau et al.02a].

Les agents sont aussi classés, selon leur mobilité. Il s'agit de leur capacité à parcourir les réseaux informatiques (réseaux locaux ou internet), et de migrer d'une station vers une autre pour exécuter un code qu'il transporte. De ce fait, on peut avoir des agents statiques ou mobiles.

En plus, les agents peuvent être classés selon leur degré d'autonomie, d'adaptation, ou de coopération : propriétés intrinsèques des agents. On distingue alors les classes suivantes :

- Agent Collaborants : Pour réaliser leurs tâches ces agents mettent l'emphase sur l'autonomie et la coopération sans avoir recours à l'adaptation et à l'apprentissage. Ces agents sont souvent utilisés pour l'interconnexion de plusieurs systèmes déjà existant ou pour résoudre des problèmes distribués. Ainsi en collaborant et en associant leurs diverses compétences individuelles ces agents peuvent résoudre un problème global dont la difficulté dépasse les compétences de chacun. Ce domaine d'application est souvent appelé **intelligence artificielle distribuée**.
- Agents Interfaces : Leurs tâches principales étant d'accompagner et d'assister l'utilisateur, ces agents mettent l'emphase sur l'autonomie et l'apprentissage. Ce sont des assistants personnels qui collaborent et communiquent avec l'utilisateur. Cette collaboration permet de mieux orienter l'exécution des différentes tâches déléguées. Ces agents sont souvent utilisés dans le domaine des **interfaces Hommes/Machines**.

- Agents d'information : Ce sont des agents capables de chercher, de collecter, d'intégrer et de manipuler des informations provenant de sources de données hétérogènes et distribuées. Ils peuvent adapter leurs techniques et critères de recherche en fonction du besoin de l'utilisateur. Ces agents peuvent aussi présenter une grande autonomie. Ils sont capables en effet de lancer leurs activités de recherche ou de collecte d'information d'une manière autonome ; soit en fonction d'un raisonnement interne, soit en fonction d'un manque d'information, soit en fonction d'une nouvelle disponibilité d'information ou de source d'information [Adam00]. Ce domaine d'application est souvent appelé **génie logiciel distribué**.
- Agents Intelligents ou "smart agents" : Ce sont des agents qui présentent les trois caractéristiques. Ils sont autonomes, adaptatifs (apprenants) et coopératifs. Par ailleurs, il est à signaler, que cette intelligence est souvent associée à la seule capacité d'apprentissage [Adam00].

	autonomie	adaptation	coopération
agents collaborateurs	+	-	+
agents interfaces	-	+	+
agents d'information	+	+	-
agents intelligents	+	+	+

TAB. 2.1 – Typologie des agents

2.3 Apport des systèmes multi-agents (SMA) dans l'ingénierie des systèmes d'information

L'apport des SMA dans l'ingénierie des systèmes d'information est défini principalement par deux aspects [Mandiau et al.02b] :

- L'évolution vers un génie logiciel distribué qui tient compte de la distribution et de l'hétérogénéité des informations sur le réseau et d'un environnement global d'applications réparties. Il s'agit principalement de systèmes d'information, évolutifs et flexibles,

qui intègrent dans leurs architectures des Agents d'Information.

- L'évolution vers des systèmes d'information personnalisés, qui s'adaptent aux besoins de l'utilisateur par l'intermédiaire d'Agents Interfaces dit aussi assistants.

Pour définir les systèmes internet intelligents et les avancées en termes d'internet et de systèmes d'information, [Levy et al.00] évoque 3 groupes d'applications :

- la modélisation de l'utilisateur via la personnalisation,
- la découverte et l'analyse de sources de données à distance,
- et l'intégration de l'information.

Ce sont précisément les applications des agents dans les systèmes d'information. Le premier point concerne la personnalisation. Les deux derniers points concernent les agents d'information pour la découverte, la recherche et l'intégration de l'information. Nous nous proposons dans la suite d'étudier les différents cas d'utilisation des agents d'information, et des agents interfaces respectivement par rapport à l'intégration de l'information dans un environnement d'information distribué, et par rapport à la personnalisation de l'information. Nous étudions ensuite les organisations typiques des systèmes d'information à base d'agents, dits aussi, dans la littérature, "systèmes d'information orientés agents" [Giorgini et al.03] ou "systèmes d'information à base d'agents" ou "systèmes multi-agents d'information" [Mandiau et al.02b] ou encore "systèmes d'informations coopératifs" [Jarras et al.02].

2.3.1 Agents d'Information pour un environnement d'information distribué

La typologie des agents d'Information utilisés dans littérature peut être présentée comme suit :

2.3.1.1 Agents Sources d'Information (Adaptateurs ou "Wrappers")

Ce sont les agents qui permettent de récupérer des données dans des sources d'informations hétérogènes et distribuées sur le réseau. Il peuvent communiquer avec une source de données à distance pour extraire des informations. Il sont dits "wrappers" [Shi02] [Wang et al.03] ou adaptateurs, puisqu'ils permettent d'adapter le contenu et le format des informations récupérées sur les sources distantes, de leur format d'origine, au format requis par l'application

SMA qui les utilisent. Ce sont des traducteurs qui permettent à un système d'information orienté agents de communiquer avec des sources de données hétérogènes ; ils sont capables de traduire les données récupérées en langages SMA et vice versa.

Comparés à des robots, ces agents logiciel sont souvent appelés *robosofts* ou *softbot*. En effet ces agents automatisent le processus d'accès à distance aux différentes sources de données. Ces sources d'informations peuvent être des bases de données distantes, des sites web sur internet, des applications sur des serveurs distants, ou même des service web.

Pour accéder à ces sources de données, chaque agent doit maîtriser le protocole de communication de la source de données dont il est responsable. Ainsi on peut avoir :

- Des agents de bases de données : il sont capables de communiquer avec des SGBD à distance.
- Agents Internet : ces agents sont capables de récupérer des informations sur des pages web ou de remplir des formulaires web d'une manière automatisée. Certains utilisent la norme Z39.50 SRW et SRU (search, and web retrieval pour récupérer les données sur le web ou Search and retrieve URL Service pour remplir à distance les formulaires web). Cette norme, appelée aussi ISO 23950, permet à ces agents non seulement la récupération des données mais aussi la recherche de ces sources.

D'autres agents, sont construits sur la base de logiciels tels que "Automate de Unisyn"³, ou "visual web task"⁴. Ces logiciels permettent d'automatiser les différentes tâches de navigation (consultation, sauvegarde, ..) qu'un utilisateur peut effectuer sur le web de façon à créer différents scénarios de navigation possibles sur internet [Kamoun et al.03].

- Des agents qui permettent l'accès à des web services. [Greenwood et al.05] ont en effet intégré le concept web service dans les systèmes multi-agents via la plate-forme Jade. Ainsi il devient possible à un agent de consulter un web service depuis son environnement SMA. Il est aussi possible à un web service de faire appel à la compétence d'un agent. Cette approche est très intéressante, dans la mesure où elle permet l'ouverture des systèmes multi-agents d'information sur les services web, aujourd'hui définis comme un standard d'échange et d'intégration d'information entre applications réparties (1.5.1.4).

L'apport des agents web services par rapport aux agents d'accès aux bases de données,

³www.unisyn.com/automate/

⁴www.sharewareriver.com/products/5428.htm

est en effet cette capacité à invoquer des méthodes et des traitements à distance.

Les agents sources d'information ou wrappers sont couramment appelés agents d'information [Sycara et al.96] puisqu'ils en sont les fournisseurs. Nous retenons cette terminologie pour la suite de l'étude.

2.3.1.2 Agents mobiles

Un agent mobile est un agent qui part d'un serveur de départ avec un ensemble prédéfini de tâches à exécuter sur différents serveurs du réseau (§ 2.2.5). Son itinéraire dans le réseau peut être prédéfini dès son départ, ou complété sur chaque serveur hôte.

Ces agents sont souvent associés à la recherche et la collecte de l'information sur plusieurs sources d'informations distribuées sur le réseau [Fan et al.99], si ces tâches prédéfinies sont un ensemble de requêtes à exécuter. Ces sources d'information peuvent aussi bien être des bases de données distribuées [Papavastrou et al.00] que des pages web sur Internet [Hammer et al.00].

Dans les environnements riches en sources d'information, ces agents présentent l'avantage de diminuer le nombre de messages envoyés sur le réseau, et d'améliorer ainsi le trafic. En effet pour récupérer des informations sur n sources de données, avec le modèle client/serveur, on a besoin de $2n$ transactions sur le réseau (requêtes/réponses). Pour récupérer les mêmes informations avec un agent mobile, il suffit de $n+1$ transactions sur le réseau : nombre de migrations de cet agent de son départ jusqu'à son retour au serveur client, et ce en passant par les n serveurs d'information.

Par ailleurs pour améliorer le temps de réponse des agents mobiles, les recherches actuelles s'intéressent à l'optimisation de leurs chemins de parcours sur le réseau [Baek et al.06] [Zgaya et al.06].

2.3.1.3 Agents Médiateurs

Les agents Médiateurs sont aussi appelés des agents de médiation de sources de données. En effet ils fournissent à ces sources des mécanismes d'interopérabilité. Ces mécanismes comprennent en plus du protocole de communication, des ontologies décrivant les données contenues dans ces sources [Roze03].

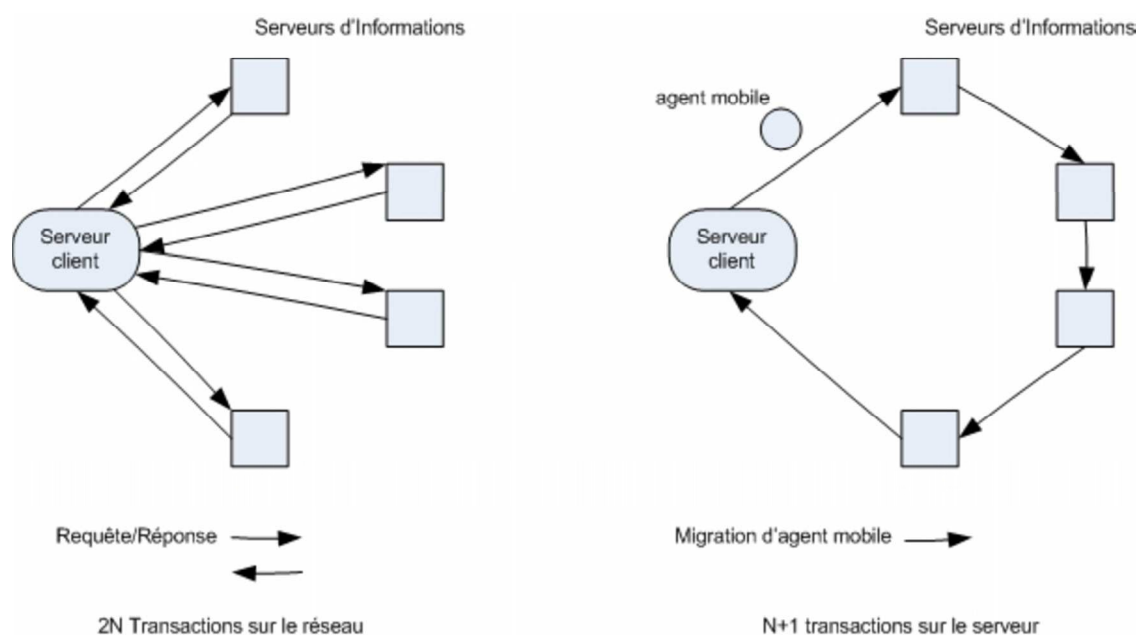


FIG. 2.8 – Collecte d'informations avec un agent mobile

[Wiederhold92] a introduit le concept de médiateur, ou manager de requêtes, entre bases de données hétérogènes et distribuées. Le médiateur qui reçoit une requête de l'utilisateur, est capable de la décomposer en sous-requêtes qu'il envoie ensuite, aux bases de données correspondantes généralement hétérogènes et distribuées. Il assure ainsi la médiation entre l'utilisateur ou l'application qui envoie la requête globale et les différentes sources de données. Ceci suppose, que le médiateur connaisse, non seulement le domaine du contenu (domaine de connaissances) de ces sources mais aussi leurs adresses sur le réseau.

Ces agents sont des facilitateurs, qui facilitent l'accès à plusieurs sources d'information. Ils sont aussi utilisés, dans les systèmes d'information orientés agents, pour la médiation, entre un utilisateur ou un agent interface et plusieurs agents "wrappers" sources d'informations. Dans un environnement multi SMA, ils peuvent aussi assurer l'interopérabilité entre plateformes multi-agents [Shi02]. Dans ce cas, ces plateformes deviennent les fournisseurs de services.

Les agents de médiation sont aussi appelés, selon [Sycara et al.96], des agents d'exécution de tâches : "Taskagent", puisqu'ils vont coordonner la répartition et l'exécution des requêtes sur les agents sources d'informations. [Jarras et al.02] et [Shakshuki et al.03] les appellent aussi "Agents courtier" ou "broker agents", puisqu'ils assurent le courtage d'information

entre clients et fournisseurs.

Leur fonction de médiation est très semblable à celle des plate-formes EAI (§1.5.1). En effet, [Jarras et al.02] résument les principales compétences de ces agents comme suit :

- associer au mieux les besoins fournisseurs et les services des fournisseurs,
- intégrer et probablement filtrer et traiter les réponses des utilisateurs pour résultats appropriés,
- aviser les utilisateurs des éventuels changement d’information.

Enfin par le courtage, la médiation, la découverte, le filtrage et l’intégration, ces agents assurent la fonction de recherche d’information dans sa globalité.

2.3.1.4 Agents Annuaire

Les agents annuaires permettent d’aiguiller les agents médiateurs dans leurs recherche d’information. Or, dans la littérature il existe trois sortes d’agents médiateurs :

- Des agents médiateurs qui possèdent, chacun, une connaissance complète sur les domaines de connaissances et les adresses réseaux de tous les agents sources d’information présents dans le système multi-agent d’information globale.
- Des agents médiateurs à connaissances limitées qui ne reconnaissent que quelques agents ressources d’information. Par conséquent, il ne peuvent adresser les sous requêtes qu’à ces agents connus à l’avance ; ceci limite leurs domaine de recherche d’information.
- Dans les systèmes multi-agents à grand nombre de sources d’information, il est difficile pour un agent médiateur de disposer d’une connaissance complète sur tous les agents sources d’information. Dans ce cas, il est nécessaire de disposer d’un agent annuaire (Directory Agent) qui disposerait d’une vision globale sur les domaines de connaissances et les adresses de tous les agents sources de données de l’environnement. Cet agent permet d’orienter et d’aiguiller les activités et les sous requêtes des agents médiateurs, et ce selon les requêtes que ces derniers reçoivent [Knoblock et al.97]. Néanmoins, les agents médiateurs gardent leurs compétences d’interrogation, de collecte et d’intégration d’informations.

Pour les deux premiers cas d’agents médiateurs, qui disposent à l’avance d’une connaissance globale sur l’ensemble des agents d’information, nous parlerons de médiation statique. Pour les agents médiateurs nécessitant un annuaire, il s’agit d’une médiation dynamique. En effet,

ces agents ne connaissent pas, à l'avance, les agents sources qu'il devraient interroger. De plus le domaine de recherche change d'une requête utilisateur à une autre.

2.3.1.5 Agents Coordinateurs

Ces agents coordinateurs ou agents superviseurs vont coordonner les tâches des agents sources d'information et des agents médiateurs. Ils peuvent aussi assurer la planification des itinéraires des agents mobiles sur un réseau informatique. Leur rôle est aussi généralement associé à des agents annuaires, ou à des agents médiateurs qui disposent d'une connaissance complète sur les autres agents présents dans leur environnement.

2.3.2 Agents Assistants pour une information personnalisée

Actuellement, nous assistons à une émergence continue et même incontrôlable des systèmes d'information, de leurs fonctionnalités et de leurs services. Un utilisateur qui cherche une information se sent, de ce fait, "noyé" dans une masse d'informations pourtant disponible. Les agents de recherche d'information permettent à des systèmes d'information d'accélérer et de rendre plus efficace la recherche sur des sources hétérogènes et distribuées pour répondre à la requête d'un utilisateur. Les agents assistants sont des agents interfaces (§2.2.5), qui en communiquant avec l'utilisateur, permettent de mieux formuler les requêtes et par conséquent de mieux orienter dès le départ le domaine de recherche.

Les agents assistants sont, en effet, généralement en communication directe avec des agents de recherche d'information. Ils leur fournissent des requêtes mieux adaptées aux besoins des utilisateurs. A leur tour, les agents de recherche d'informations, vont leur fournir un ensemble de résultats de recherche, que ces derniers peuvent encore filtrer et personnaliser. Assimilé à un assistant personnel, chaque agent interface est souvent au service d'un seul utilisateur.

Les assistants sont des agents adaptatifs qui ont une grande capacité d'apprentissage. En se basant sur une observation et une mémorisation du comportement de l'utilisateur face au système, de l'historique des requêtes de ce dernier, et de l'ensemble des préférences exigées à chaque connexion, ces agents arrivent à intégrer dans leurs bases de connaissance un modèle de l'utilisateur [Mandiau et al.02b].

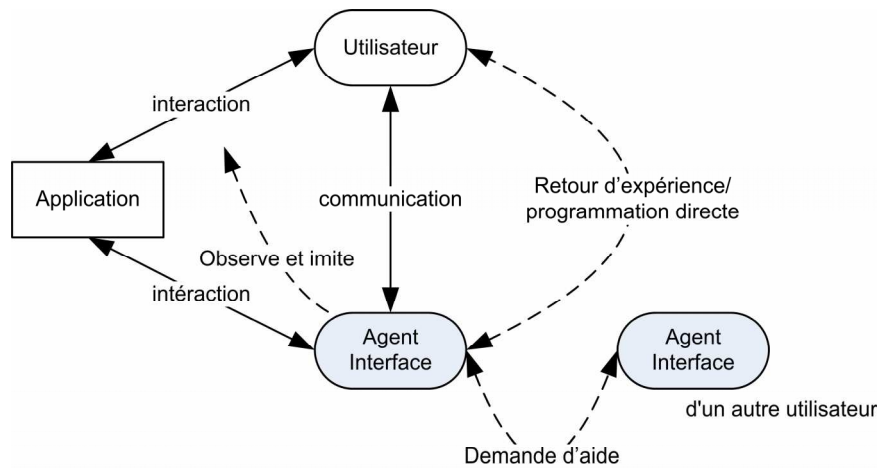


FIG. 2.9 – Fonctionnement d'un agent interface assistant

[Maes94] définit quatre modes d'apprentissage :

- en observant et en imitant l'interaction de l'utilisateur avec l'application,
- en recevant des réponses positives et/ou négatives de l'utilisateur,
- en recevant explicitement des instructions de l'utilisateur,
- ou en demandant des conseils à d'autres agents.

Il peuvent ainsi, grâce à cet apprentissage, anticiper la formulation de la requête de l'utilisateur et même de la recherche, en proposant des préférences, des critères de recherche et mêmes des résultats de recherche relatifs au profil enregistré. Nous citons, à titre d'exemple, le système Laetizia créé au MIT par [Lieberman97], qui consiste en un assistant personnel pour la navigation sur le web. En fonction, de l'historique des pages web visitées par l'utilisateur, ce système propose d'autres pages susceptibles de l'intéresser. WebMAte [Chen et al.98] est un autre assistant personnel pour l'aide à la navigation sur le web, et qui intègre aussi des compétences de recherche d'informations.

2.3.3 Organisation types des Systèmes d'information orientés agents

Dans les systèmes d'information orientés agents, pour assurer les fonctionnalités de personnalisation, de recherche et d'intégration de l'information, deux organisations multi-agents sont souvent proposées :

2.3.3.1 Organisations du type Interfaces-Médiateurs-Adaptateurs

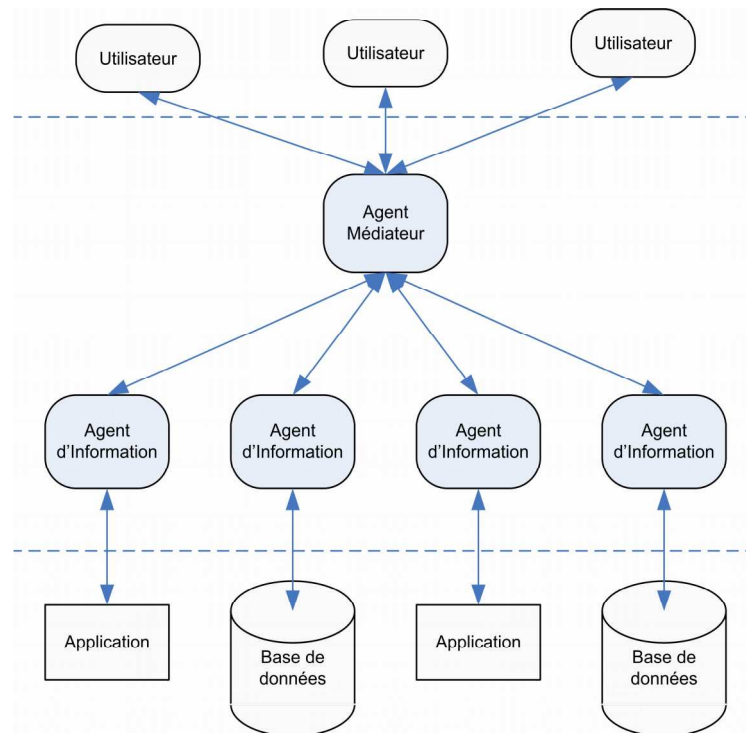


FIG. 2.10 – Architecture médiateur-Adaptateurs [Shi02]

Ces types d'organisation impliquent :

- Des agents Interfaces pour la personnalisation des requêtes et des réponses, et l'agentification des interactions Homme/Machine selon le modèle présenté précédemment (Fig 2.9).
- Des agents Médiateurs pour la recherche de l'information, la répartition et la décomposition d'une requête globale en plusieurs sous-requêtes à envoyer à des agents d'information. Ils assurent aussi l'intégration des différentes réponses reçues pour composer une réponse globale.
- Des agents d'information ou adaptateurs, qui vont adapter les sous requêtes aux différentes sources d'information qu'ils gèrent. Ils adaptent aussi les réponses reçues de ces sources au langage adopté dans l'environnement SMA.

Bien que très semblables, ces organisations présentent des différences en termes de structures organisationnelles et des interactions mises en oeuvre entre les différents agents. Nous

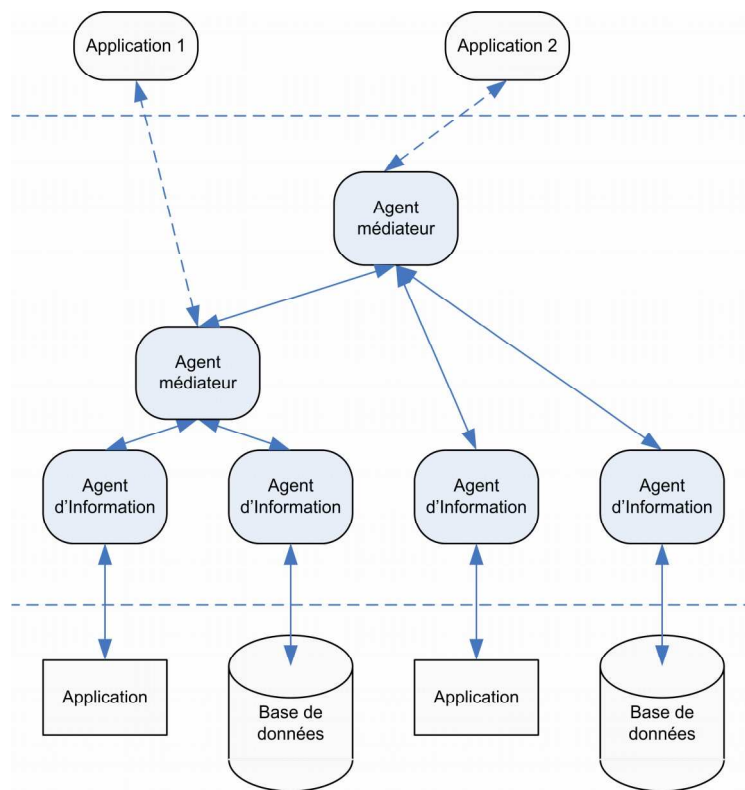


FIG. 2.11 – Architecture médiateurs-Adaptateurs du TSMMS [Shi02]

distinguons :

Les organisations Médiateurs-Adaptateurs Ce sont des architectures qui ne font appel qu'à des agents médiateurs et des agents d'information. Leur objectif étant la recherche et l'intégration d'information, il sont souvent appelés intergiciels ou "middelwares" orientés agents.

Nous distinguons dans cette catégorie deux organisations :

- une organisation à base d'un seul agent médiateur, qui joue le rôle d'un routeur de requêtes (Fig 2.10) [Shi02]. Ceci suppose qu'il dispose d'une connaissance complète sur les domaines des agents d'information qu'il gère, ainsi que leurs adresses sur le réseau. Les requêtes étant gérées par un seul agent médiateur, une file d'attente et un goulot d'étranglement se créent au niveau de leur traitement.
- Une organisation à base de plusieurs agents médiateurs, où chaque agent médiateur

gèrent un ensemble prédéfini d'agent d'informations (Fig 2.11), et ce selon le besoin de l'application qui lui fait appel.

Dans certaines organisations, des agents médiateurs peuvent considérer d'autres médiateurs comme agents d'informations. Ainsi ils délèguent la médiation sur une partie du domaine global de recherche (l'ensemble des agents d'information de l'environnement), à des agents médiateurs intermédiaires.

Ce type d'organisation est appliqué à l'EAI pour l'intégration d'applications hétérogènes et distribuées. Nous citons à titre d'exemples le système TSIMMIS (The Stanford-IBM Manager of Multiple Information Sources), qui constitue une plate forme d'intégration d'applications (Fig 2.11 et Fig 1.5) et le système MIWeb plate forme d'intégration de sources web [Busse et al.04]. De nos jours, les applications "Peer to Peer" se basent aussi sur de telle organisations multi-agents [Panti et al.02]. En effet chaque PC peut être considéré comme médiateur quand il télécharge des informations sur plusieurs stations distantes pour ensuite les intégrer. Mais il est aussi considéré comme agent source d'informations quand il est sollicité lui-même pour un téléchargement.

Les organisations Interfaces-Médiateurs-Adaptateurs Par rapport aux architectures précédentes qui ne gèrent que l'intégration de l'information, celles-ci apportent une personnalisation de l'information par une agentification de l'interaction Homme/machine. Ainsi, entre l'utilisateur et l'agent médiateur, s'interface un assistant ; il s'agit d'un agent interface. A chaque utilisateur on associe un agent interface [Martin et al.97], [Busse et al.04], [Park et al.05]. Cet agent l'aide à formuler ses requêtes, à définir ses préférences, pour les transmettre ensuite au médiateur(Fig 2.12).

D'autres organisations comme celles présentées par [Sycara et al.96], associent à chaque utilisateur non seulement un agent interface mais aussi un agent médiateur. Ceci permet d'éviter les goulots d'étranglement au niveau du traitement puisque ce dernier est distribué sur plusieurs agents. Chaque médiateur traite les requêtes de son utilisateur.

Par ailleurs dans les environnements riches en information, ces agents médiateurs peuvent avoir recours à un agent annuaire pour savoir vers quels agents d'information s'adresser. Ainsi le domaine de recherche de chaque agent médiateur est redéfini d'une manière dynamique par l'annuaire, et ce pour chaque requête de l'utilisateur. Il s'agit d'une médiation dynamique.

Néanmoins, ces organisations se composent aussi de médiateurs dits statiques dont les

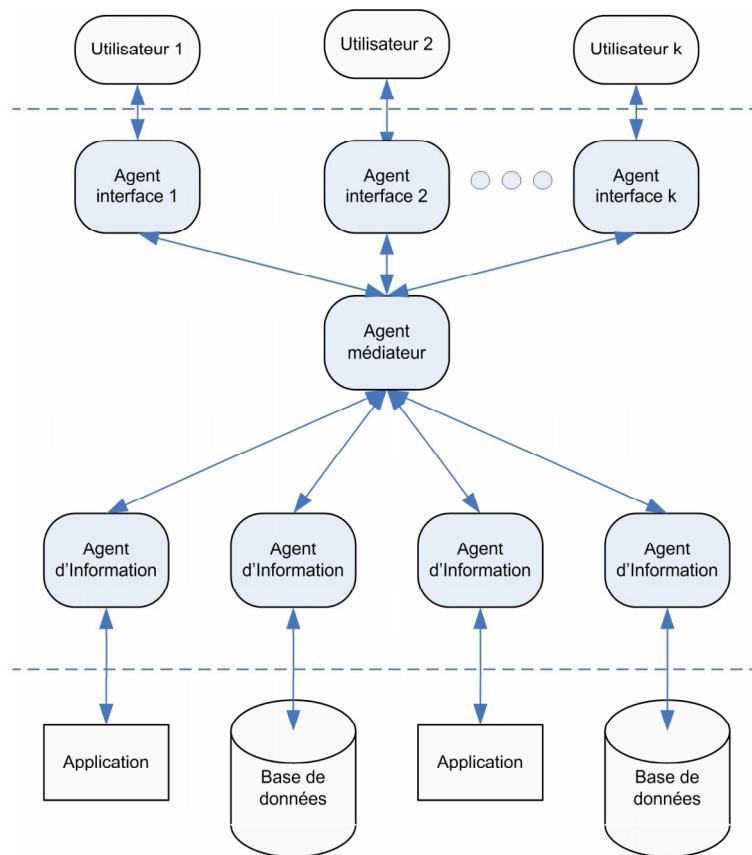


FIG. 2.12 – Architecture Interfaces-médiateur-Adaptateurs

domaines de recherche sont prédéfinis à l'avance. Ils sont des supérieurs hiérarchiques de plusieurs agents d'information. Mais ils sont considérés par les agents médiateur dynamiques comme de simples agents d'information.

Pour définir une organisation généralisée du type Interfaces-Médiateurs-Adaptateurs, qui tient compte des différents aspects présentés précédemment, nous proposons l'organisation illustrée par la Figure 2.13.

L'organisation Interfaces-Médiateurs-Adaptateurs est un modèle structurel pour plusieurs applications. Parmi ces applications, nous citons :

- Infosleuth : C'est un système multi-agents pour la recherche coopérative d'informations dans un environnement de bases de données distribuées. Il a été appliqué à des domaines médicaux [Nodine et al.99].
- NetSA (Networked Software Agents) : C'est aussi un système de recherche d'informa-

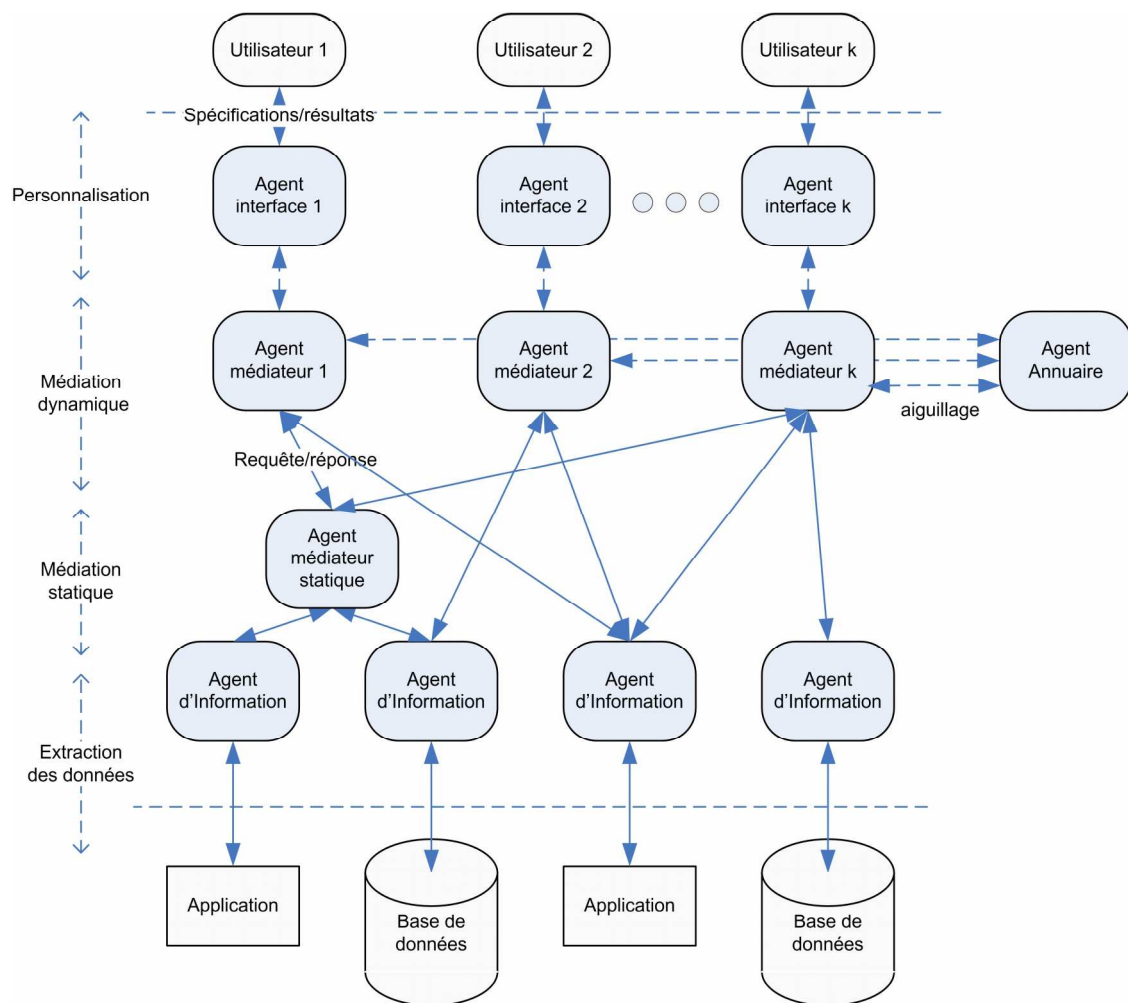


FIG. 2.13 – Architecture Interfaces-Médiateurs-Adaptateurs

tions mais dédié aux environnements riches en information [Cote et al.98].

- UMDL : c'est un système de recherche coopérative de documents dans des bibliothèques numériques [Vidal et al.98].

Tous ces systèmes sont qualifiés par [Jarras et al.02], de systèmes d'information coopératifs, qui en coopérant assurent une recherche et une intégration d'informations.

2.3.3.2 Organisation SMA à base d'agents mobiles

Les systèmes d'information orientés agents et à base d'agent mobiles, sont très utilisés dans les environnements riches en sources d'information, puisque grâce à leur migration, ces

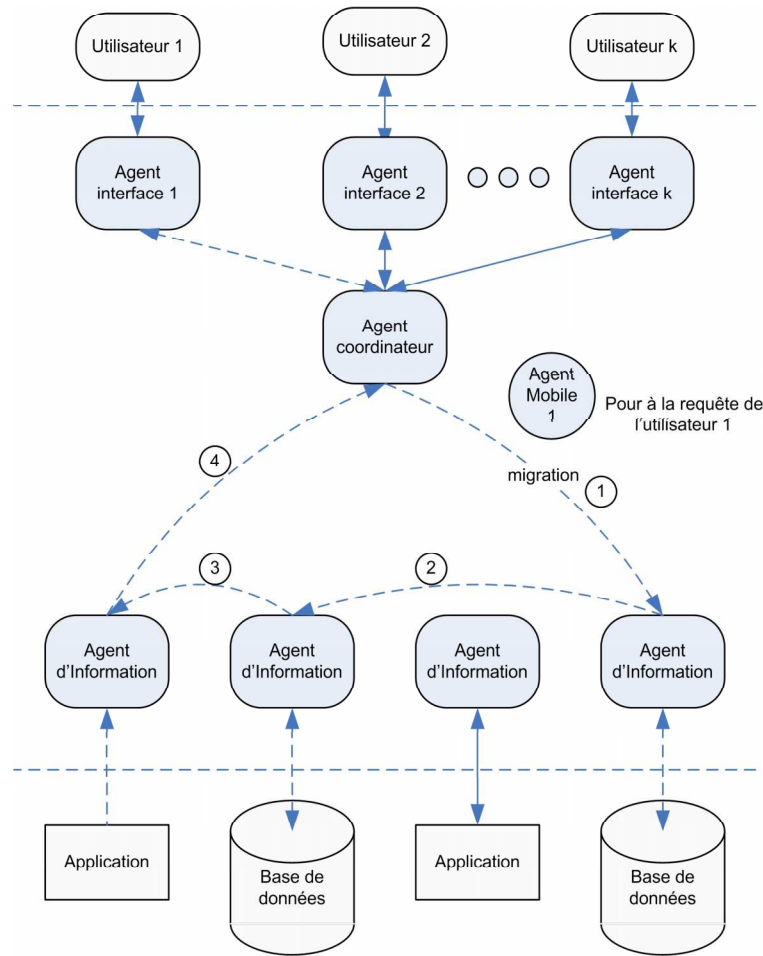


FIG. 2.14 – Architecture Interfaces-Coordinateur-Agents Mobiles-Adaptateurs

derniers permettent d'améliorer le trafic sur le réseau en diminuant le nombre de transactions [Bergamachi et al.01], [Yang et al.02]. Ces organisations présentent des agents interface et des agents adaptateurs, respectivement pour la communication avec les utilisateurs et les sources d'informations. Par rapport aux architectures précédentes qui adoptent des médiateurs pour la recherche d'information, celles ci utilisent les agents mobiles. Ces agents mobiles sont souvent gérés par d'autres agents [Bennani et al.02]. Deux fonctions nécessaires sont associées à ces derniers :

- La coordination ou la planification, qui consiste à planifier selon le besoin de l'information, l'itinéraire à parcourir sur le réseau.
- Et la fonction d'annuaire qui consiste à répertorier l'ensemble des serveurs d'information disponibles dans l'environnement et à gérer le parc global des agents mobiles.

2.3.4 Applications des SMA dans les systèmes d'information dédiés au transport

Les systèmes multi-agents sont abondamment utilisés dans le domaine du transport, principalement dans la simulation et la modélisation du trafic routier ou ferroviaire ; et dans la résolution de problèmes d'optimisation difficile relatifs au transport.

Leurs applications dans les systèmes d'information dédiés au transport, restent limitées, et concernent principalement :

- la conception et la réalisation des systèmes d'aide à l'exploitation (SAE voir §1.4.1).
- La conception et la réalisation des systèmes d'information d'aide au déplacement.

2.3.4.1 Applications dans les systèmes d'aide à l'exploitation (SAE)

Pour les systèmes d'information d'aide à l'exploitation, le principal apport des organisations multi-agents réside dans l'intégration de l'information provenant des différents services de l'exploitant [Balbo et al.04], et dans l'apport de solutions viables à des problèmes distribués de régulation de la qualité et de la fréquence des services en cas de perturbations [Fayech03], [Laichour02].

2.3.4.2 Applications dans les systèmes d'aide au déplacement

Pour les systèmes d'information d'aide au déplacement, les applications des SMA sont multiples. Dans la littérature, ces systèmes d'information orientés agents sont communément appelés ATIS (Advanced Traveler Information Systems)[Yin et al.05][Lin et al.03] ou PTA (Personal Travel Assistance) [Steiner97][Dieterich et al.97].

En effet, l'information voyageur est par définition une information composée. C'est une agrégation de données provenant de sources hétérogènes et distribuées que l'utilisateur ne connaît pas forcément à l'avance. Ces données peuvent concerner la description de l'itinéraire lui même ; dans ce cas elles sont multi-exploitants. Mais elle peuvent aussi concerner des informations annexes relatives aussi à différents fournisseurs de services.

De plus, cette information est toujours relative à un voyageur. Ce dernier planifie son voyage par rapport à ses propres besoins, critères et préférences.

De ce fait, le paradigme des systèmes multi-agents d'information, qui assurent personnalisation, collecte et intégration de l'information, a trouvé une place naturelle dans ce domaine de l'information voyageur.

Mais encore, l'application des agents pour la personnalisation, la collecte et l'intégration de l'information voyageur, devient aujourd'hui pour plusieurs études sur la conception des SMA, un exemple typique pour la validation et l'implémentation [Dam et al.03a].

Nous étudions dans la suite, les organisations proposées dans la littérature pour les systèmes multi-agents de l'information voyageur, et les apports de la personnalisation de cette information.

Organisations multi-agents pour l'intégration de l'information voyageur Dans une optique de standardisation, d'interopérabilité entre plate-formes multi-agents, et pour montrer de surcroît la faisabilité des technologies orientées agents, la FIPA (Foundation of Intelligent Physical Agents)⁵ propose une organisation multi-agents, du type Interfaces-Médiateurs-Adaptateurs, pour implémenter un système d'information voyageur.

Le système proposé permet à un utilisateur de réserver son voyage simplement en indiquant le détail de sa requête à un agent interface appelé PTA (Personal Travel Agent), qui prendra ses critères et ses préférences et les transmettra à un agent médiateur TBA (Travel Broker Agent). Ce dernier va décomposer le voyage en une série de segments. Ce médiateur va ensuite interroger un ensemble d'agents adaptateurs sources d'informations dits TSA (Travel Service Agent) qui représentent les fournisseurs de services (compagnie aérienne, de bateaux, hôtels...). Les informations sur les segments du voyage sont retournés au TBA qui va les agréger et les concaténer pour envoyer l'offre assemblée au PTA. Cet agent interface va contrôler la validité de cette offre par rapport aux critères de l'utilisateur. Si l'offre est susceptible d'être reçue, elle est présentée à l'utilisateur qui procédera à une réservation. Dans le cas échéant, une nouvelle recherche est lancée.

Cette organisation automatise et personnalise le processus de réservation de voyage. Le système créé est une agence de voyage virtuelle.

D'autres travaux de recherche Allemands comme ceux de [Dieterich et al.97] ou [Steiner97]

⁵www.fipa.org

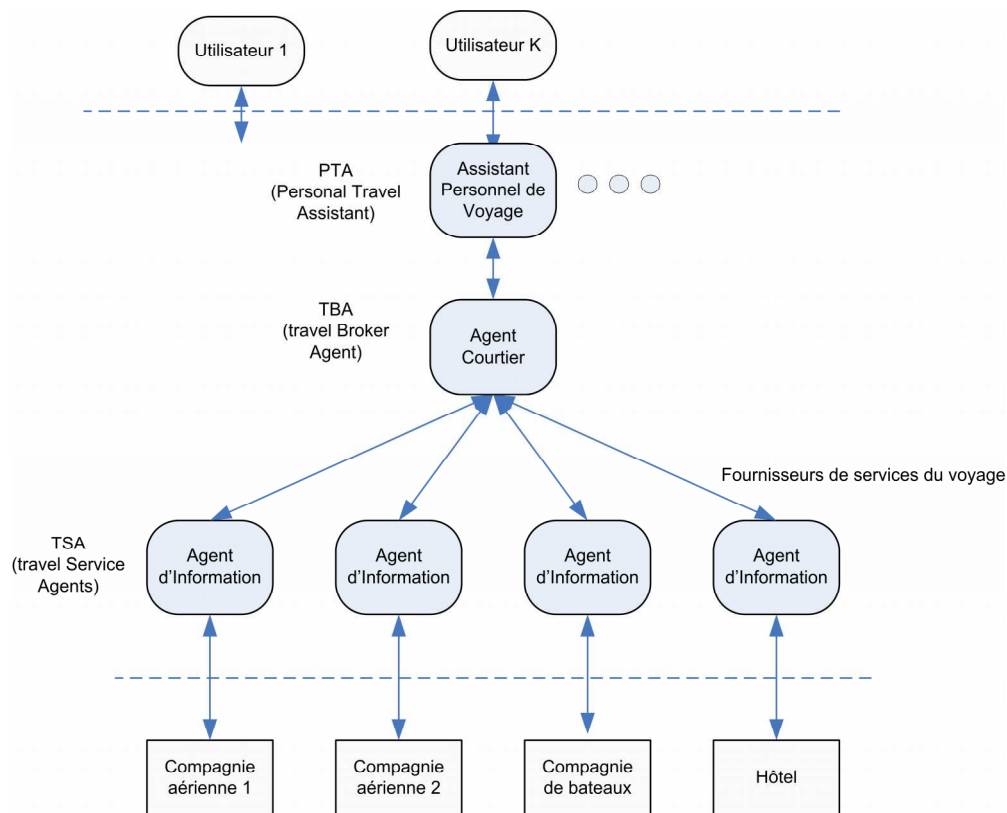


FIG. 2.15 – Organisation FIPA pour un système multi-agents d'information voyageur

et Hollandais comme ceux des Universités Technologiques de Delft et de Eindhoven ⁶, reprennent la même organisation Interfaces-Médiateurs-Adaptateurs présentée par la FIPA pour concevoir d'autres systèmes multi-agents d'information voyageur. En plus de la planification d'itinéraires par agrégation de segments de voyages, ces systèmes proposent une information annexe sur des services d'accompagnement du voyage (météo,...). Ils proposent aussi un processus de suivi et de monitoring. Ce dernier permet de détecter toute condition ou perturbation qui empêcherait l'utilisateur de réaliser son voyage conformément au plan fourni au départ par le PTA (Personal Travel Assistant). Ainsi en cas de perturbation, ces systèmes peuvent proposer en temps réel, d'autres itinéraires alternatifs aux voyageurs. L'incident est détecté par le fournisseur du service perturbé, qui va en informer les médiateurs qui l'ont interrogé. Ces derniers vont par la suite informer les agents interfaces des utilisateurs concernés.

⁶www.pita.tudelft.nl

Pour que l'utilisateur puisse accéder à son agent assistant pendant son voyage, mais aussi pour que cet agent puisse communiquer avec son utilisateur, pendant ce voyage, l'informer d'une éventuelle perturbation et lui proposer des itinéraires alternatifs; les travaux précédents, introduisent le concept d'agent assistant personnel de voyage (PTA) ubiquitaire. Cet agent est alors implémenté sur des terminaux mobiles (Téléphone portable, assistant ou autre...) et fait appel à distance à des agents médiateurs TBA (Travel broker Agents) ou agents courtiers qui accomplissent leurs tâches sur des serveurs distants.

[Yin et al.05] propose un autre système multi-agent d'information voyageur appelé SCATEAgent (Context-Aware Software Agents for Multimodal Travel) basé aussi sur le même type d'organisation Interfaces-Médiateurs-Adaptateurs. Associant à chaque utilisateur un agent assistant personnel qui cumule la fonction d'interface et celle de la médiation, ce système assure aussi la fonction de monitoring et de suivi de l'information.

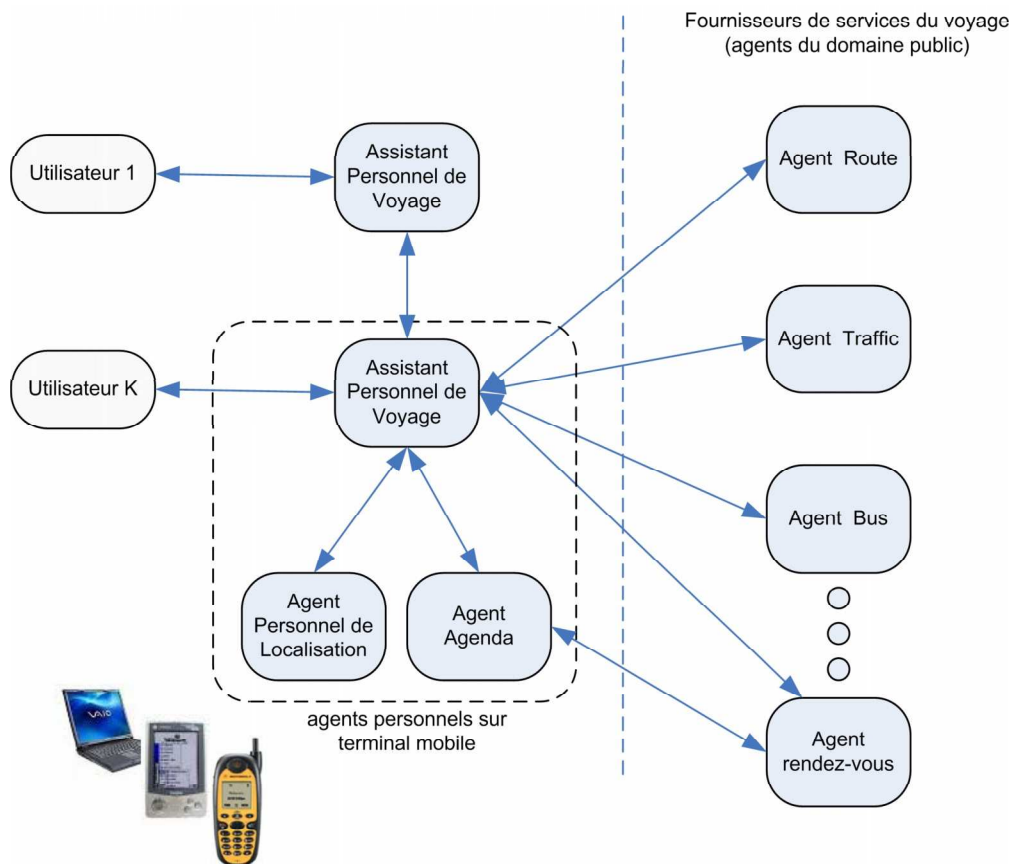


FIG. 2.16 – Organisation SMA du système SCATEAgent [Yin et al.05]

Mais à la différence des systèmes précédents, il réalise ce monitoring en tenant compte de la localisation du voyageur. En effet, il introduit un Agent Personnel de Localisation (APL) qui sera intégré avec l'agent assistant personnel de voyage dans le terminal mobile de l'utilisateur. Pendant le voyage, l'APL va déterminer de manière continue la localisation du voyageur pour la transmettre ensuite à l'agent assistant de voyage. Ayant cette localisation, ce dernier sera capable de chercher les services annexes de proximité pour les proposer ensuite à l'utilisateur. En cas de perturbation, l'agent assistant de voyage sera aussi en mesure de proposer les itinéraires alternatifs, en fonction de la localisation du voyageur.

Par ailleurs, [Lin et al.03] propose une organisation à base d'agents mobiles pour les systèmes multi-agents d'information voyageur.

D'autres travaux, reprennent les organisations précédentes et les transposent au transport multimodal dans les agglomérations [Strahan et al.03], [Mehandjiev et al.04], [Camacho et al.00], [Yin et al.05]. Ils proposent ainsi des systèmes multi-agent d'information pour l'aide au déplacement multimodal.

Organisations multi-agents pour la personnalisation de l'information voyageur

Les travaux de RoZé [Roze et al.04], [Roze et al.05], [Roze et al.00] et ceux d'Anli [Anli et al.04], [Anli06]; proposent une organisation multi-agents pour la personnalisation appliquée à l'information voyageur. MAPIS (Multi-Agent Personalized Information System) est une organisation composée de quatre modèles d'agents cognitifs qui correspondent aux principales activités réalisées par le système :

- La gestion des interactions homme-machine ou "assistance" est assurée par des agents assistants. Leur rôle consiste à comprendre les requêtes de chaque utilisateur et de les aider à mieux définir leurs besoins.
- La gestion des données et de leurs recherche : Il s'agit d'agent mobiles pour la recherche et la collecte des données.
- La gestion des profils des utilisateurs : Ceci consiste à créer et à maintenir une base de données contenant les connaissances que le système dispose sur les utilisateurs. Les agents gestionnaires de profils sont capable d'apprendre et de mettre à jour les profils, par communication directe avec les utilisateurs, par observation de leurs interactions avec le système, ou par traitement sur les données connues telles que l'historique des

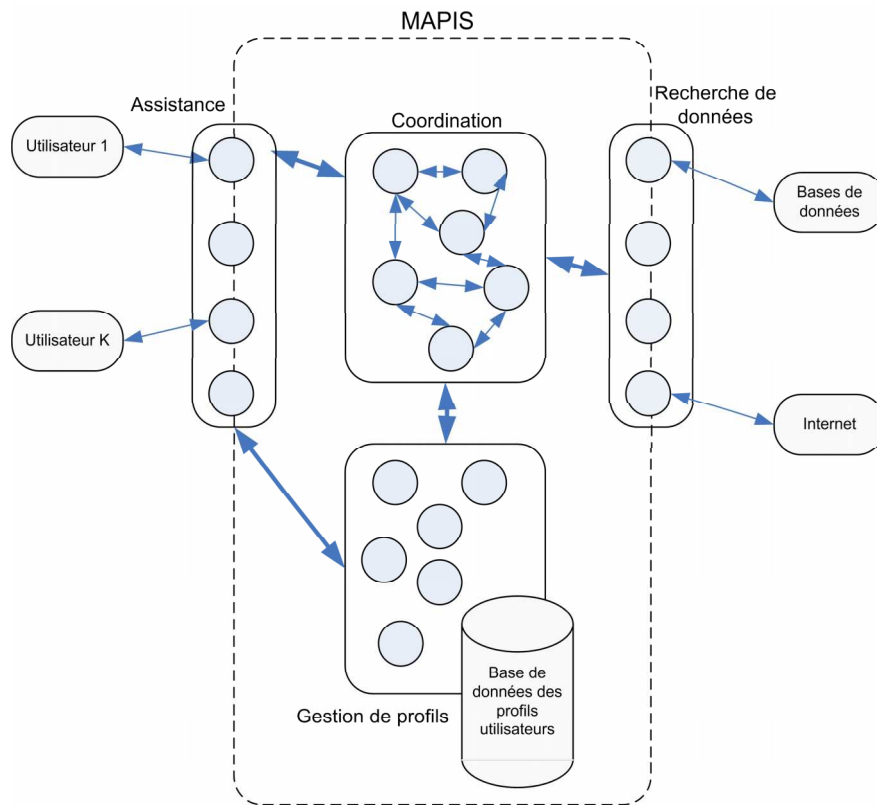


FIG. 2.17 – Organisation MAPIS [Roze et al.04]

interactions.

- La coordination et la gestion des flux et des traitements des données : Cette tâche est assurée par des agents coordinateurs qui vont superviser l'ensemble des activités et des interactions des autres agents. Ces agents reçoivent un ensemble d'offres d'itinéraires des agents de recherche d'information. Ils procèdent ensuite à un filtrage qui consiste à déterminer la meilleure solution, parmi l'ensemble reçu, qui s'adapte le mieux non seulement au contexte lié à la demande (heure de départ, heure d'arrivée, station de départ et station d'arrivée) mais aussi au profil de l'utilisateur.

2.4 Nouvelle OMA d'aide au déplacement multimodal

En résumé, les services fournis par les systèmes multi-agents d'information dans le domaine de l'information voyageur sont :

- La planification par la composition et l'intégration d'itinéraires multimodaux et inter-

modaux provenant de sources hétérogènes et distribuées.

- Le suivi de l'information pour informer, en cas de perturbation de services, les usagers concernés et leur proposer des itinéraires alternatifs.
- L'ubiquité en permettant aux usagers du transport d'accéder, à distance et pendant leurs trajets, aux systèmes d'information à partir de dispositifs mobiles (Téléphone mobile, PDA, ...).
- La personnalisation par l'agentification de l'interaction Homme/machine avec les système d'information en considérant les profils des utilisateurs.

Les deux derniers points présentent, certes, une valeur ajoutée non négligeable, mais il s'agit de concepts qui restent liés aux systèmes multi-agent d'information d'une façon générale. Ils peuvent par conséquent être adaptés à des domaines autres que le transport. Les deux premiers points, constituent une valeur ajoutée, qui reste par contre spécifique au domaine du transport multimodal et multi-opérateurs. En effet la recherche et la composition d'itinéraires, requièrent une organisation d'agents adaptée au contexte actuel et à la cartographie des systèmes d'information de transport existants (§ 1.4.2). De plus la recherche et la composition d'itinéraires requièrent des méthodologies et des algorithmes d'optimisation distribués spécifiques au domaine.

De ce fait, nous nous proposons dans la suite une organisation multi-agents pour l'aide au déplacement multimodal et multi-opérateurs, qui tient compte des spécificités et du contexte de ce domaine et qui assure principalement les fonctions de recherche et de composition d'itinéraires.

Il s'agit de concevoir un système multi-agent d'information pour l'aide au déplacement multimodal. Ce système permettra à plusieurs systèmes d'informations monomodaux ou mono-opérateurs de coopérer pour construire une offre de mobilité globale sur un territoire prédéfini. Nous l'appelons Système d'information Coopératif de Mobilité ou SICM.

Le SICM est un intergiciel (middleware) qui sera un client parmi d'autres usagers des systèmes d'information monomodaux mono-opérateur. Il devient alors l'intermédiaire entre les différentes sources d'informations hétérogènes et distribuées d'une part, et les usagers du transport, d'autre part. Ce système doit alors être capable à la fois de trouver la bonne source d'information pour l'interroger selon les différentes requêtes des utilisateurs, de récupérer les réponses, et de regrouper les informations de manière significative pour proposer une offre d'itinéraire global optimisé.

Certes les organisations présentées précédemment, répondent intuitivement à ce besoin de conception. Mais avant de proposer une organisation typique, il convient d'étudier les méthodologies de conception des systèmes multi-agents et notamment des systèmes multi-agents d'information.

2.4.1 Méthodologies de conception des systèmes multi-agents

Plusieurs méthodologies ont été développées pour la conception des systèmes multi-agents. Elles accompagnent les développeurs essentiellement dans les phases d'analyse et de spécification des besoins et dans la phase de conception du système multi-agents. Certaines incluent aussi la phase de développement et d'implémentation.

Leurs domaines d'applications respectifs restent à ce jour souvent limités. Cette limitation est due principalement à un manque de maturité de ces méthodes. Des études bibliographiques ont essayé de répertorier ces méthodologies et de les classer en différentes catégories. On distingue d'après [Lahlouhi01] et [Laichour02] :

- Des méthodologies développées pour la conception de systèmes multi-agents très particuliers. Celles-ci n'atteignent pas, en général, un degré de maturité qui leur permet d'intégrer l'ensemble des caractéristiques des systèmes multi-agents. De ce fait elles ne sont pas utilisées dans des domaines d'applications SMA, autre que celui pour lequel elles ont été conçues.
- Des méthodologies qui sont générales mais spécifiques aux SMA. Elles sont encore à un stade de développement.
- Celles qui sont des extensions des méthodologies orientées objets.
- Celles qui sont des extensions des méthodologies de développement des systèmes à base de connaissances.

Les méthodologies les plus utilisées sont les extensions des systèmes à base de connaissances et les extensions des méthodologies orientées objets [Wood et al.00]. Les extensions des systèmes à base de connaissance telles que DESIRE ou MAS-CommonKADS [Iglesias et al.98], fournissent des techniques qui permettent de modéliser principalement les connaissances des agents. Le processus qu'ils présentent n'inclut pas la phase d'implémentation des agents.

Par ailleurs, les méthodologies de conception qui sont des extensions des technologies

orientées objets telles que MASE [Deloach et al.01], GAIA [Wooldridge et al.00], Prometheus [Padgham et al.02], MESSAGE/UML [Leal et al.01], AUML (Agent-Unified Modeling Language [Bauer et al.01], TROPOS [Bresciani et al.04], restent à ce jour les plus utilisées. Elles couvrent les phases d'analyse et de conception et certaines, telles que MASE ou Prometheus, couvrent via des outils spécifiques la phase d'implémentation [Dam et al.03b].

Le principal avantage des méthodologies à extension d'objets est que l'expérience, la popularité et le succès des méthodologies orientées objets sont transposés dans le domaine des agents. Ainsi, il est plus facile pour les adeptes de l'orientée objet d'appréhender et d'intégrer la technologie Agent [Laichour02].

Nous constatons aussi que la majorité des applications SMA sont implémentées en langages orientés objet, sur des plate-formes développées elle-même, en langages orientés objet telles que JAVA ou C++. Certes, il est aussi plus facile de passer d'une méthodologie d'analyse et de conception d'agents qui s'inspire de l'objet, à un développement en langages orientés objet.

Ces méthodologies laissent apparaître des approches entre l'objet et l'agent qui sont :

- Un agent est un objet actif ayant un état mental. Il est de plus autonome : il a le choix de dire "non" à un éventuel appel de ses méthodes ou compétences. Il est aussi souvent porté par une thread ou processus parallèle.
- Objets et agents échangent des messages pour communiquer.
- Des diagrammes UML (Langage de modélisation orienté objet) tels que les diagrammes d'activité, de collaboration, de séquence ou de déploiement sont souvent utilisés pour modéliser les systèmes multi-agents.

De ce fait, nous assistons actuellement, à une émergence de travaux qui visent à concevoir des systèmes multi-agents associés à divers domaines d'application, notamment le domaine des systèmes d'information pour le transport [Laichour02] [Roze03] [Fayech03]; sans pour autant adopter une méthodologie de conception prédéfinie dans la littérature. S'inspirant des méthodologies agents orientés objets, ces travaux abordent le problème de conception comme un problème de modélisation de l'organisation SMA, en utilisant différents diagrammes UML. Cette modélisation est réalisée en deux étapes :

- Une description locale des agents : Il s’agit de définir les différents types d’agents à impliquer, en précisant leurs rôles dans l’organisation. Cette description aborde aussi les comportements propres des différents agents, et les différents scénarios d’activités qu’il peuvent exécuter. Elle est souvent associée à des diagrammes UML d’activités pour les agents impliqués.
- Une description sociale des agents qui décrit le comportement global du système via des scénarios d’interactions entre agents, illustrés généralement par des diagrammes de collaboration ou par des diagrammes de séquences. Cette description présente l’organisation globale du système multi-agents.

Pour concevoir le SICM, Système d’Information Coopératif de Mobilité, nous optons pour cette alternative. En effet, nous allons adopter une organisation qui existe dans la littérature. Notre apport sera de l’adapter à la recherche et à la composition d’itinéraires dans le contexte actuel des systèmes d’information d’aide au déplacements.

2.4.2 Du contexte transport vers une OMA d’aide au déplacement multimodal

Nous rappelons que dans le contexte actuel des transports en France et dans le monde, la cartographie des systèmes d’information multimodale sur un territoire donné, est une juxtaposition de systèmes d’information d’aide au déplacement souvent monomodaux et mono-opérateur (§1.4.2). En effet, à l’image des réseaux de transport aux quels ils sont associés, ces SIAD gèrent des stations d’intersection relatives aux pôles d’échange. Ainsi pour composer un itinéraire multi-opérateurs, l’utilisateur doit connaître à l’avance :

- les SIAD à impliquer dans cette recherche,
- les pôles d’échange par lesquels il souhaite passer, pour consulter ensuite les offres des tronçons d’itinéraires entre ces différentes stations d’intersection, sur les SIAD retenus.

Il va ensuite concaténer les tronçons d’itinéraires pour construire un itinéraire global qui correspond à ses propres critères de recherche. Si nous analysons cette démarche de recherche et de composition d’itinéraires effectuée par l’utilisateur, nous constatons qu’il réalise deux types de raisonnement :

- un premier raisonnement qui lui permet de déterminer les SIAD à interroger,
- un deuxième raisonnement qui lui permet de composer l'offre de l'itinéraire global. Il s'agit de réaliser différentes combinaisons de tronçons d'itinéraires, pour constituer différentes propositions d'une offre globale.

Les systèmes d'information multimodale distribués tels que DELFI ou EU-SPIRIT, ont essayé d'automatiser cette démarche de recherche d'itinéraires. En effet, comme chaque utilisateur, ces systèmes utilisent des annuaires qui fournissent les informations requises sur les SIAD et les pôles d'échanges : connaissances nécessaires à cette démarche de composition. Se basant sur ces connaissances, ces systèmes appliquent, comme l'utilisateur, deux raisonnements :

- Un premier raisonnement qui leur permet de déterminer les SIAD à interroger. Dans ce cas, ce raisonnement consiste à déterminer le SIAD associé à la station départ de l'itinéraire demandé, et celui associé à la station d'arrivée. Pour compléter la chaîne de SIAD qui permettent de constituer l'offre globale, ces systèmes font toujours appel à un seul SIAD intermédiaire. En effet le système DELFI permet de relier deux SIAD régionaux via le SIAD national. Le système EU-SPIRIT permet de relier deux SIAD nationaux via un SIAD international. L'offre globale sera toujours une concaténation deux tronçons d'itinéraires de courte distance et d'un tronçon d'itinéraire relativement de longue distance, sans considérer les autres combinaisons d'opérateurs possibles.
- Un deuxième raisonnement leur permet de composer l'offre de l'itinéraire global. Il s'agit de réaliser différentes combinaisons de tronçons d'itinéraires, pour constituer différentes propositions d'une offre globale, en considérant les différents pôles d'échange.

Or dans le contexte actuel des transports publics, un déplacement intermodal et multimodal n'est pas forcément un déplacement inter-régional. Ainsi nous pouvons avoir dans la même agglomération plusieurs opérateurs de transports, ce qui implique la possibilité d'avoir plusieurs SIAD intermédiaires entre un SIAD associé à une station de départ et une autre associée à une station d'arrivée. D'où le besoin de nouveaux raisonnements qui permettent de déterminer le domaine de recherche et d'assurer la composition de l'offre globale.

Nous déterminons ici deux fonctionnalités qui seront associées aux agents du SICM pour la recherche et la composition d'itinéraire. Le deuxième apport du SICM serait cette capacité d'accompagner l'utilisateur en cas de perturbation du service d'un opérateur pour proposer des itinéraires alternatifs.

2.4.2.1 Organisation du SICM

Pour concevoir le SICM, nous commençons par associer à chaque SIAD d'un opérateur de transport, et par conséquent, à chaque réseau de transport un Agent d'Information (AI). L'AI est un extracteur d'information à partir du SIAD. Pour ce faire, l'AI dispose en interne de l'adresse réseau (internet ou autre) de ce SIAD, et des protocoles de communication qui lui permettront de lui envoyer des requêtes et d'en recevoir des réponses.

Un AI, comme le SIAD auquel il est associé, est capable de répondre à une requête d'itinéraire, du type aller de A vers B, si A et B sont des stations gérées par le même agent.

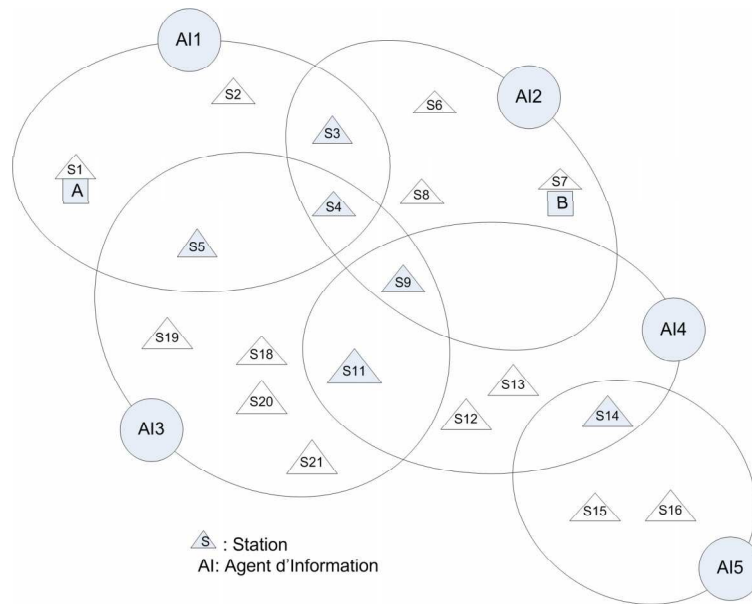


FIG. 2.18 – Agents d'Information

La composition et la recherche d'itinéraires seront affectées à des agents médiateurs. Nous les appelons dans ce contexte, agents courtiers compositeurs (AC). Un AC est capable de composer une offre globale d'itinéraire en interrogeant différents AI et en récupérant différentes offres locales.

L'AC ne dispose pas d'une vue globale de tous les AI du SICM. En effet pour répondre à une requête donnée, l'AC n'est pas capable de déterminer un domaine de recherche, d'où l'intérêt d'un Agent Annuaire Sélectionneur (AS). Ce dernier sera en effet sollicité par les médiateurs pour délimiter les domaines de recherches relatifs aux différentes requêtes. Ceci revient à leur envoyer, l'ensemble des AI sélectionnés qui définissent ce domaine et l'ensemble des stations d'intersection. Il s'agit d'une médiation dynamique puisque l'AC ne connaît pas à l'avance, pour une requête donnée, les AI à sélectionner.

Par ailleurs, nous associons ensuite à chaque utilisateur un Agent Interface Utilisateur (AIU) qui permettra à ce dernier de formuler sa requête et de recevoir les résultats. Pour distribuer le traitement des requêtes des utilisateurs, on associe chaque AIU à un médiateur AC.

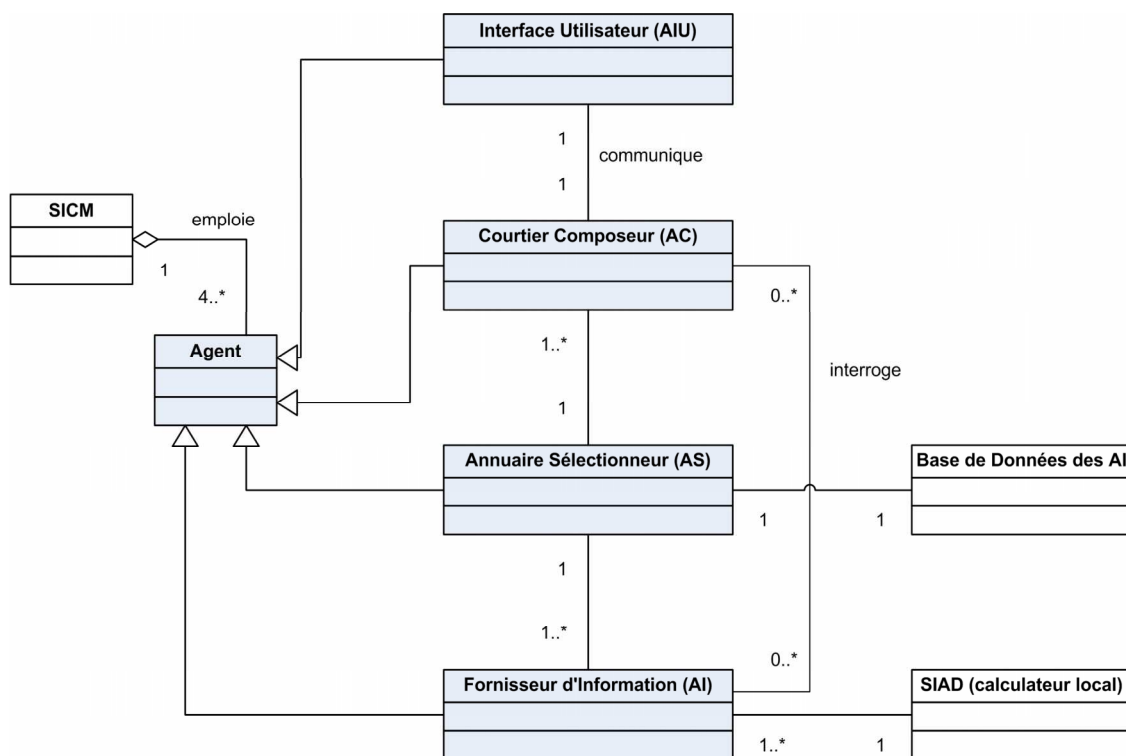


FIG. 2.19 – Agents du SICM

En conclusion, nous reprenons l'organisation Interfaces-Médiateurs-Adaptateurs (§2.3.3.1) illustrée par le diagramme de classes de la figure 2.19.

2.4.2.2 Agents du SICM

Suite à cette description du SICM qui a permis, à partir du contexte du transport public, de déterminer les rôles et les responsabilités des agents à impliquer dans l'organisation SMA, nous abordons dans la suite la description du comportement propre de chaque agent. Cette description est illustrée par des diagrammes UML d'activités qui permettent de retracer leurs différents scénarios de fonctionnement. A la différence des objets, ces agents sont portés par des processus légers ("threads"). Cette propriété, dite aussi continuité, leur permet d'être soit en activité, soit à l'écoute de leur environnement, et en attente d'un stimulus qui déclencherait une autre activité. Il s'agit par conséquent de cycles d'activités qui reviennent toujours à une phase initiale d'attente.

Comportement de l'agent d'information (AI) L'AI est à l'écoute de deux types de stimulus :

- Soit il reçoit une requête de l'AC qu'il devra transformer pour l'envoyer à son SIAD. Dans ce cas, il va transmettre à cet AC soit la réponse à sa requête, soit un refus dans le cas où l'information requise n'est pas fournie par le SIAD.
- Soit il reçoit une alerte sur la perturbation d'un service de transport. Dans ce cas il va transmettre cette alerte aux AC qui ont fait appel à des informations concernant ce service.

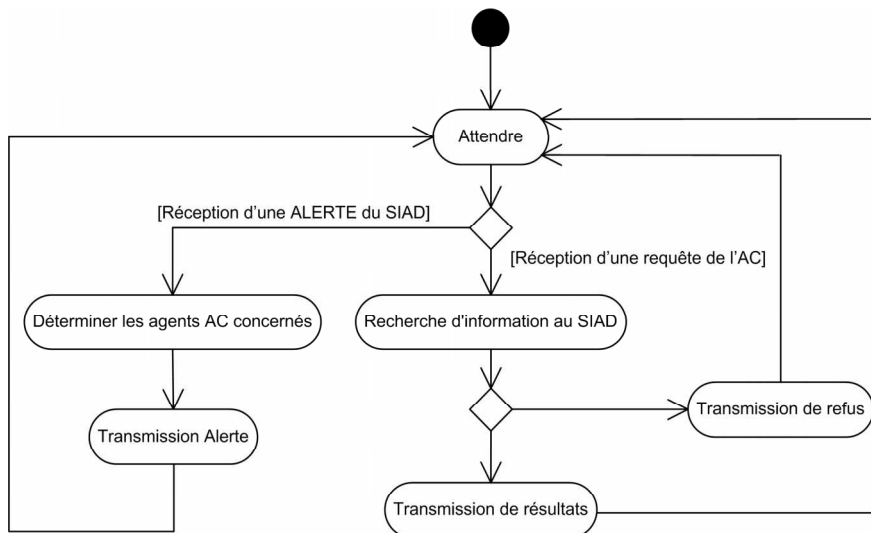


FIG. 2.20 – Comportement de l'Agent d'Information (AI)

Comportement de l'agent courtier (AC) L'agent courtier compositeur (AC) présente deux scénarios d'activités :

- Soit il reçoit une requête de l'AIU. Dans ce cas, il demande à l'agent annuaire sélectionner un domaine de recherche. Si cette demande aboutit, il commence une recherche et une composition d'itinéraire avec les AI sélectionnés, sinon un refus est transmis à l'AIU pour l'inciter à modifier sa requête. Par la suite, si cette composition aboutit, l'AC transmet les résultats à l'AIU, sinon un refus est aussi transmis à l'AIU.
- Par ailleurs si l'AC reçoit une alerte de perturbation de service de l'un des AI qui ont été retenus pour la réponse relative à une requête précédente, il transmet à son tour cette alerte à l'AIU.

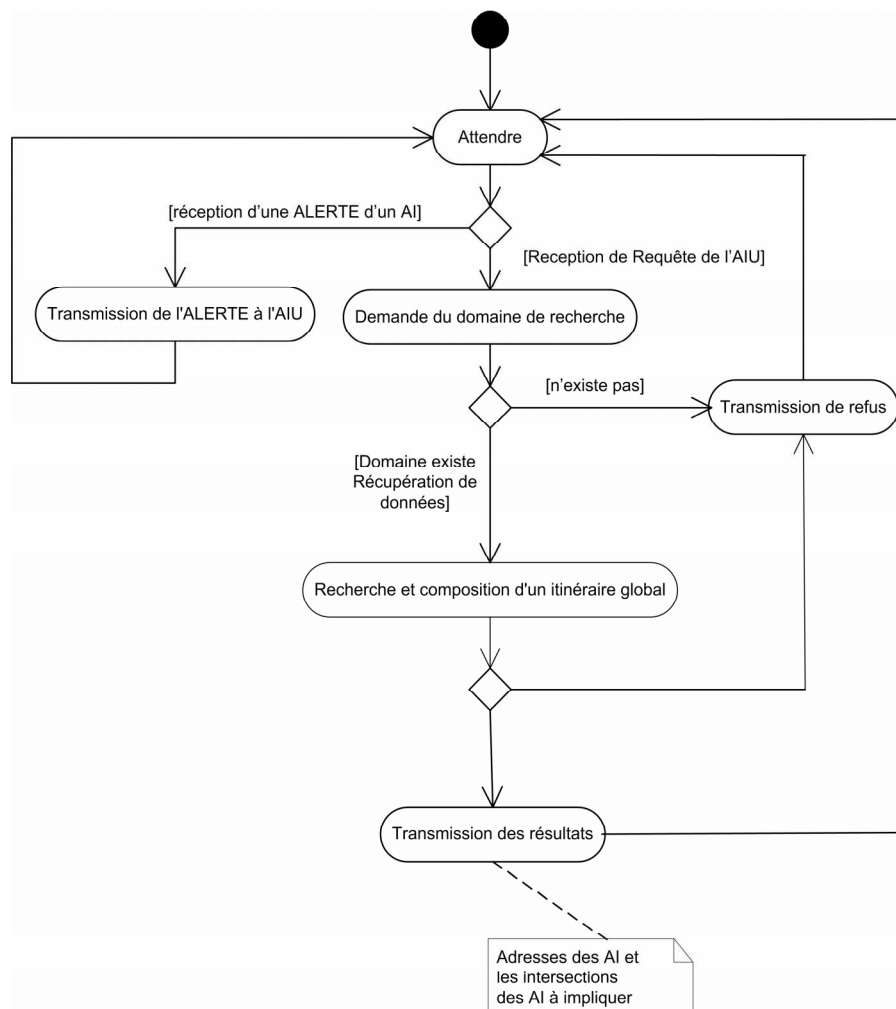


FIG. 2.21 – Comportement d'Agent Courtier (AC)

Comportement de l'agent annuaire sélectionneur (AS) L'AS présente deux scénarios de fonctionnement :

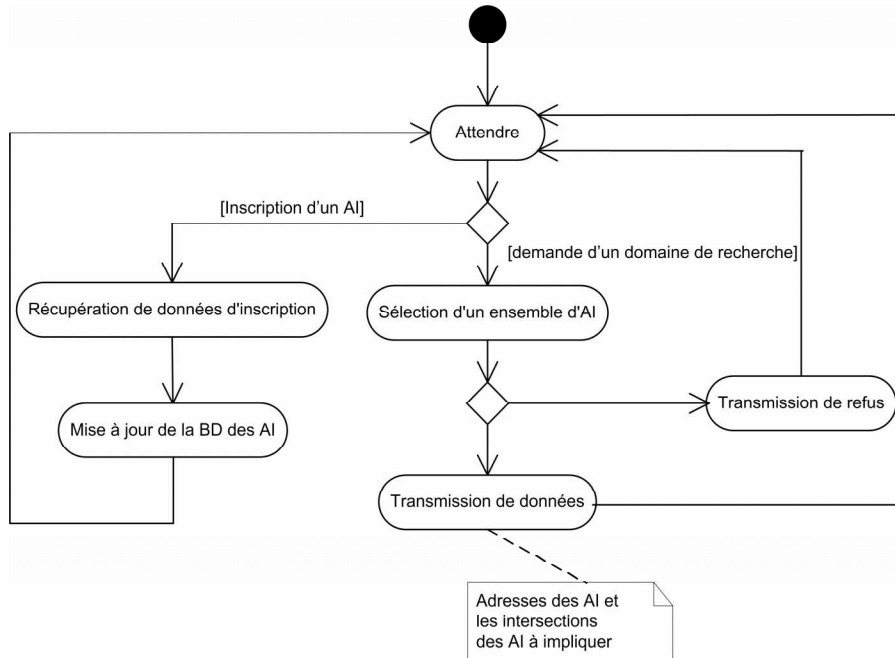


FIG. 2.22 – Comportement de l'Agent Annuaire Sélectionneur (AS)

- Soit un nouveau AI demande à s'inscrire au SICM. Il fournira alors un ensemble de données : l'ensemble des pôles d'échanges qu'il gère, ce qui permettra de déterminer les AI du SICM qui lui sont adjacents. La base des données des AI du SICM est mise à jour.
- Soit il reçoit d'un agent compositeur (AC), une demande de domaine de recherche. Cette requête lui fournit la station de départ et la station d'arrivée. Il détermine alors les AI de départ et d'arrivée associés à ces stations, et sélectionne un ensemble d'AI qui permet de relier les deux premiers. S'il arrive à constituer le domaine de recherche relatif à cette requête, il transmet le résultat à l'AC, sinon un refus est transmis à l'AC qui incitera l'AIU à reformuler sa requête.

Comportement de l'agent interface utilisateur (AIU) L'AIU permet à l'utilisateur de formuler sa requête et de préciser ses critères de recherche. En effet quand il reçoit une requête de l'utilisateur, ce dernier la transmet à l'AC, et attend sa réponse. S'il s'agit d'un refus, il va demander à l'utilisateur de reformuler sa requête, sinon il lui présente les résultats

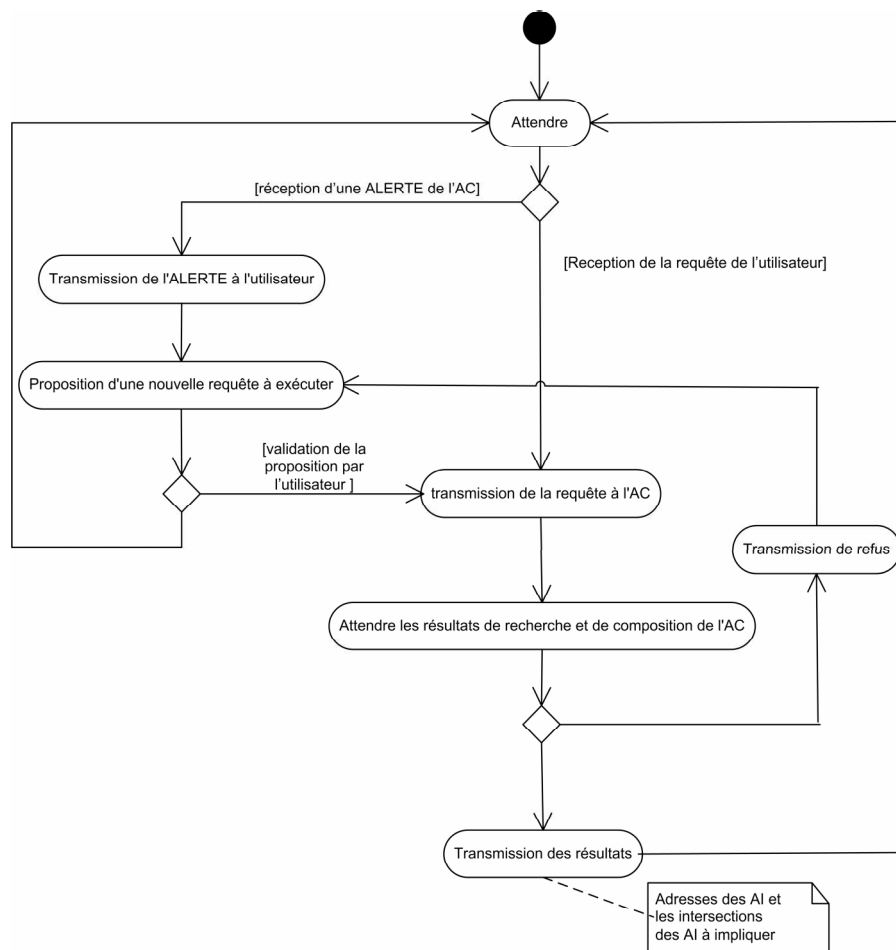


FIG. 2.23 – Comportement de l'Agent Interface Utilisateur(AIU)

de la recherche.

Par ailleurs, s'il reçoit une alerte de perturbation provenant de l'AC, il va proposer à l'utilisateur une nouvelle requête. Si celle-ci est validée, il la transmet de nouveau à l'AC pour recommencer le processus de recherche et de composition d'itinéraires.

2.4.2.3 Comportement global du SICM

Le SICM permet de rechercher et de composer un itinéraire global sur un ensemble de SIAD distribués et hétérogènes. Il permet aussi d'alerter l'utilisateur en cas de perturbation de service. Cette deuxième fonctionnalité revient à proposer des itinéraires alternatifs. Il s'agit de recommencer la recherche et la composition avec d'autres critères ou sur un nouveau domaine de recherche.

Nous nous intéressons dans la suite au scénario de recherche et de composition d'un itinéraire global. Ce scénario est illustré par le diagramme de séquences UML suivant. Ce diagramme détaille les communications et les interactions entre agent qui, à partir d'une requête de l'AIU, vont aboutir à une offre globale.

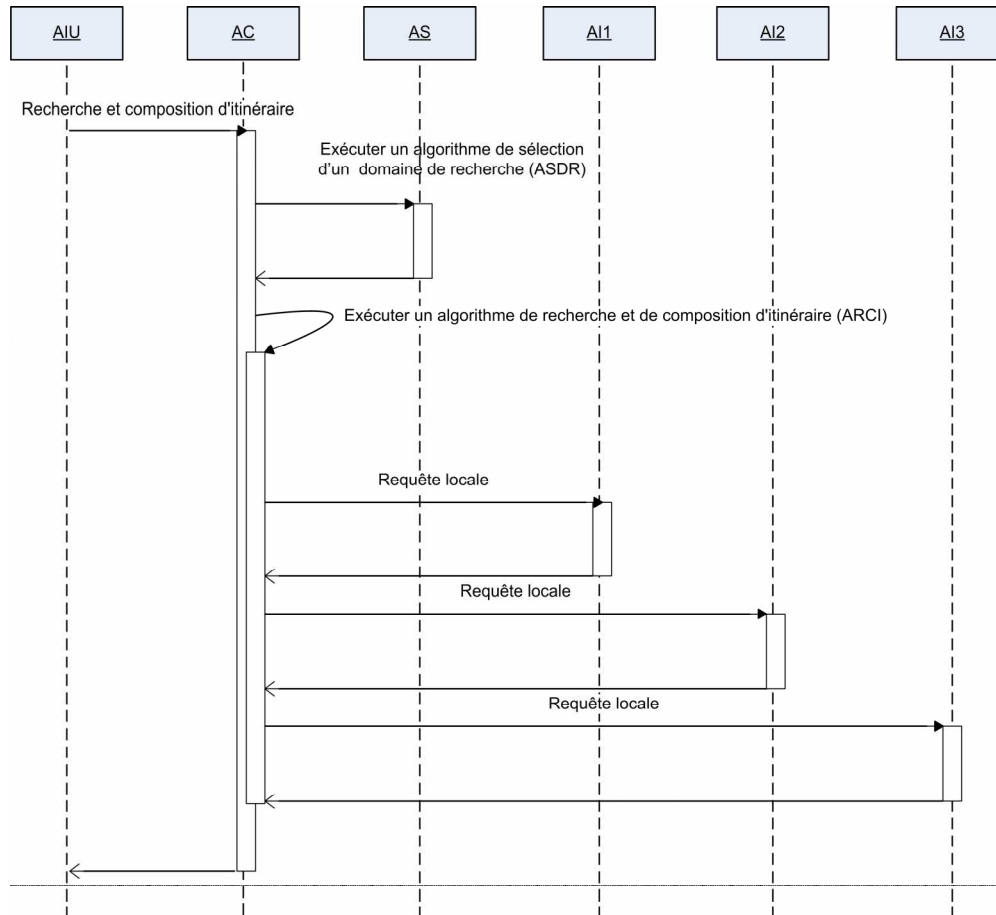


FIG. 2.24 – Comportement global du SICM

2.5 Conclusion

L'étude de l'apport des systèmes multi-agents dans les systèmes d'information nous a permis de recenser dans la littérature plusieurs organisations SMA dédiées aux systèmes multi-agent d'information et notamment celles dédiées aux systèmes multi-agents de l'information voyageur.

L'apport de l'approche multi-agents à l'information voyageur en termes de fonctionnalités,

a été résumé en quatre points : la personnalisation de l'information, la mobilité, le suivi de l'information en cas de perturbation et la planification des itinéraires. Nous nous sommes intéressés à la planification d'itinéraires, et au suivi de l'information, particulièrement dans un contexte de transport public multimodal et multi-opérateur.

Dans ce contexte, le SICM proposé est un intergiciel capable de rechercher et de composer l'information multimodale dans un environnement distribué et hétérogène.

Deux algorithmes sont nécessaires pour son fonctionnement : un algorithme pour la sélection d'un domaine de recherche (ASDR) exécuté par l'agent annuaire (AS) et un algorithme de recherche et de composition d'itinéraires (ARCI) exécuté par les agent compositeurs (AC). Ces deux algorithmes seront développés dans le chapitre 3. A ce stade cette conception reste préliminaire, puisque le contenu détaillé des messages échangés entre les différents agents, n'a pas été défini. Ce détail sera fourni après l'étude des deux algorithmes.

En définitif le SICM, permet comme les systèmes d'information multimodale distribués, d'exploiter les SIAD existants, et de laisser la responsabilité de mise à jour de l'information aux opérateurs de transport. Par ailleurs le SICM se présente comme un système ouvert capable d'inclure des AI d'une manière automatisée dans le domaine global de recherche, ou de les exclure. Cette approche permet aussi d'intégrer ultérieurement des aspects de personnalisation ou d'ubiquité.

Chapitre 3

Algorithmes de recherche et de composition d'itinéraires

3.1 Introduction

La conception préliminaire du SICM, système d'information coopératif de mobilité, nous a permis de déceler deux besoins d'optimisation. Une première optimisation qui permettrait, par rapport à une requête utilisateur, de limiter le domaine de recherche et de le définir par un ensemble d'agents d'information. Une deuxième optimisation consiste à rechercher et à composer le meilleur itinéraire global, correspondant à la même requête en interrogeant les différents AI retenus.

Le premier problème revient à trouver une chaîne minimale, ou des chaînes minimales d'AI adjacents qui permet ou permettent de relier l'AI associé à la station de départ et celui associé à la station d'arrivée, de façon à pouvoir par la suite composer un itinéraire global en interrogeant les AI d'une chaîne. Une fois le domaine de recherche fixé, le deuxième problème revient à trouver sur le domaine retenu l'itinéraire global optimal correspondant à la requête de l'utilisateur et à ses critères de recherche. Ce qui revient à trouver un plus court chemin dans un environnement distribué.

Pour répondre à ces besoins, nous commençons par un état de l'art sur les algorithmes de plus court chemin dans le domaine de transport et notamment ceux appliqués à la planification des itinéraires multimodaux. Ce qui nous permettra par la suite de choisir, d'adapter ou de proposer des algorithmes qui répondent aux deux besoins du SICM définis précédemment.

3.2 Algorithmes de plus court chemin dans le domaine du transport

Les algorithmes de plus court chemin sont abondamment utilisés dans le domaine du transport, et notamment dans le domaine du transport public. Ils sont utilisés principalement dans deux domaines d'applications :

- l'analyse des performances des réseaux de transport. En effet lors de la conception de ces réseaux, des urbanistes vont analyser l'accessibilité de certains lieux publics moyennant ces réseaux. Cette analyse comprend une phase de modélisation du réseau, et une phase d'analyse des performances de ces systèmes. Elle permet ainsi de faire un choix sur l'emplacement des lignes de transport, des arrêts et des stations.
- l'aide au déplacement et la planification des itinéraires dans les ATIS (Advanced Traveler Information System). Ceci revient à implémenter des algorithmes de plus court chemin sur les SIAD (Système d'Information d'Aide au Déplacement), qui permettent de répondre à des requêtes locales d'itinéraires provenant de différents utilisateurs.

Dans ces applications le réseau de transport est modélisé par un graphe $G(N, E)$. Les noeuds de ce dernier, éléments de l'ensemble N , représentent les stations du réseau. Les arcs du graphe, éléments de l'ensemble E représentent les courses assurées par les moyens de transport de ce réseau pour aller d'un noeud vers un autre. Les arcs sont valués avec un poids qui représente le coût de la course. Le problème de plus court chemin consiste à trouver le chemin qui minimise le coût global, et ce en partant d'un noeud de départ a vers un noeud d'arrivée b .

3.2.1 Algorithmes du plus court chemin classiques

Ce problème est résolu par des algorithmes PCC de plus court chemin. Pour chercher le plus court chemin sur un graphe donné, ces algorithmes construisent généralement un arbre des plus courts chemins, et permettent ainsi de déterminer pour un noeud de départ donné l'ensemble des plus courts chemins vers tous les autres noeuds. Les algorithmes sont souvent classés en deux catégories [Nonato et al.99] : Les algorithmes de fixation d'étiquettes (label Setting Algorithms) tels que celui de Dijkstra [Dijkstra59] de complexité $O(|N|^2)$, et les algorithmes de correction d'étiquettes tels que les algorithmes classiques de Bellman[Bellman58],

de Ford[Ford56] ou de Moore[Moore59], qui présentent une complexité de $O(|N||E|)$.

d_{ij} : poids de l'arc (i, j) , α_{ai} : coût du chemin qui va de a vers i , $prec(i)$: noeud précédent de i ,
 $N^+(i)$: ensemble des noeuds successeurs de i , $N_c(i)$: ensemble des noeuds courants.

1. initialisation :

$$\alpha_{ai} = \infty \text{ et } prec(i) = \infty \text{ pour } \forall i \in N, i \neq a$$

$$\alpha_{aa} = 0, prec(a) = a \text{ et } N_c = \{a\}$$

2. Sélection d'un noeud :

$$\text{soit } i \text{ le noeud ayant } \min_{j \in N_c}(\alpha_{aj})$$

$$N_c = N_c \setminus \{i\}$$

3. Exploration des successeurs possibles

Pour $\forall j \in N^+(i)$ Faire :

$$\left| \begin{array}{l} \text{si } \alpha_{aj} > \alpha_{ai} + d_{ij} \text{ alors Faire} \\ \left| \begin{array}{l} \alpha_{aj} = \alpha_{ai} + d_{ij} \\ prec(j) = i \end{array} \right. \\ \text{si } j \notin N_c \text{ alors } N_c = N_c \cup \{j\} \end{array} \right.$$

4. critère de Fin : $N_c = \emptyset$ sinon retour à l'étape 2

$N_c(i)$: ensemble des noeuds courants.

FIG. 3.1 – Algorithme Dijkstra de plus court chemin

L'algorithme le plus utilisé et le plus cité à ce jour est celui de Dijkstra (voir Fig 3.1). En effet ce dernier a été adapté à différents contextes, et notamment aux réseaux à grande échelle : réseaux de transport qui présentent un nombre important de noeuds et d'arcs. Une heuristique (A*) a été développée pour orienter la recherche sur ce type de graphe [Pallottino et al.97].

Par ailleurs cet algorithme a été aussi adapté pour résoudre des problèmes de plus court chemin dans les graphes dynamiques, où le coût $d_{ij}(t)$ de l'arc (i, j) est - à la différence des graphes statiques- une fonction du temps de départ t de la station i . Plusieurs travaux, se sont intéressés à ce problème et notamment ceux de [Pallottino et al.97], [Chabini et al.98],

et [Ahuja et al.02]. Ces graphes dynamiques sont abondamment utilisés pour modéliser les réseaux de transport public, où entre deux stations, on effectue plusieurs courses avec différents temps de départ, en proposant différents temps de trajets.

D'autres travaux se sont intéressés au problème du plus court chemin sur un graphe multimodal [Bousedjra05] [Foo et al.99]. Dans ce cas plusieurs modes de transport permettent d'aller d'une station de départ vers une station d'arrivée.

D'autres variantes ont été développées pour résoudre le problème de plus court chemin multi-critères où chaque arc est caractérisé par différents coûts (durée, coût, confort...) [Climaco et al.82], [Martins84], [Martins et al.99], [Zidi06].

Enfin, d'autres variantes permettent de calculer les k plus courts chemins de i , vers j . Ces variantes sont en effet très utilisées dans le domaine des ATIS pour présenter à l'utilisateur plusieurs offres d'itinéraires possibles entre i et j , qui soient rangés selon le critère de recherche [Subramanian97], [Wu et al.04], [Chen et al.04].

3.2.2 Discussion

Nous n'avons pas la prétention d'énumérer tous les travaux relatifs aux problèmes du plus court chemin dans le domaine de transport. Dans la littérature, il existe en effet, des travaux de synthèse qui visent à mettre à jour cet état de l'art [Nonato et al.99], [Bousedjra05]. Nous survolons cet état de l'art pour chercher des algorithmes qui s'adaptent aux besoins d'optimisation précédemment définis.

Ainsi, pour trouver les chaînes minimales d'AI qui permettent de répondre à une requête d'itinéraire global, nous proposons d'utiliser les algorithmes de k plus court chemins sur un graphe qui représenterait tout l'environnement des AI. Chaque chemin représente alors une chaîne d'agents d'information capable de répondre à la requête globale. Nous expliquerons ce choix dans la suite du chapitre.

Par ailleurs, pour la composition en ligne d'un itinéraire global, rares sont les algorithmes de plus court chemin qui permettent de prendre en compte cet aspect distribué des données provenant des SIAD des différents opérateurs. En effet tous les algorithmes du plus court chemin cités à ce stade permettent de trouver le plus court chemin sur un ensemble de données centralisé. A notre connaissance, Wang et Kaempke [Wang et al.04] ont développé le premier algorithme de plus court chemin qui permet de composer un plus court chemin global en

ligne à partir de plusieurs réseaux représentés par des graphes statiques. Nous étudierons cet algorithme en détails. Nous l'adapterons par la suite à notre problématique de composition d'itinéraire global, à partir de différents réseaux de transport public représentés par des graphes dynamiques.

3.3 Algorithme de plus court chemin distribué

3.3.1 Modélisation du problème

Pour modéliser le problème du plus court chemin distribué, nous associons à chaque AI, et par conséquent à chaque SIAD, une classe C . La i^{eme} classe est notée C_i . C_i permet de résoudre le problème de plus court chemin sur un graphe $G_i(N_i, E_i)$ où N_i et E_i sont respectivement l'ensemble de noeuds et l'ensemble des arcs relatifs à la classe C_i . Un noeud représente une station relative au réseau de transport d'un opérateur. Un arc représente une course réalisée par un moyen de transport géré par le même opérateur.

Ainsi un arc $(a, b) \in E_i$ si et seulement si $a \in N_i$ et $b \in N_i$ et le transport entre a et b est assuré par un moyen de transport géré par C_i . Le poids de l'arc (a, b) est noté $d(a, b)$.

Par ailleurs $(a, b)_i$ est l'arc $(a, b) \in E_i$, E_i l'ensemble des arcs de C_i . En effet si a et b font parti de plusieurs C_i , alors on peut avoir plusieurs possibilités d'arcs (a, b) ou chaque arc $(a, b)_i$ représente un moyen de transport géré par une classe C_i .

Soit G le graphe associé à l'ensemble de l'environnement distribué défini par n classes C_i . Il est noté :

$$G(N, E) \quad N = \bigcup_{i=1}^n N_i \quad E = \bigcup_{i=1}^n E_i$$

$G(N, E)$ est une agrégation des G_i . Certes, comme les réseaux de transport qu'ils représentent, les G_i présentent des noeuds d'intersection relatifs aux stations d'intersection dites pôles d'échange, et qui permettent, au cours d'un déplacement, de changer d'opérateur de transport. Par exemple, si a et b sont des noeuds d'intersections des classes C_i et C_k telles que :

$$a \in N_i \cap N_k, b \in N_i \cap N_k, \text{ avec } (a, b)_i \in E_i \text{ et } (a, b)_k \in E_k$$

alors il y a deux arcs entre a et b dans le graphe global $G(N, E)$ de poids respectifs $d(a, b)_i$ et $d(a, b)_k$.

Un itinéraire est défini par un chemin entre deux noeuds de G . Un chemin est dit **local** si tous les sommets et les arcs qui le constituent appartiennent à une classe C_i ; il s'agit d'un chemin intra-classe. Il est dit **global** s'il inclut des sommets ou des arcs relatifs à plusieurs classes; il s'agit d'un chemin inter-classes.

On note $p_i^k(a, b)$ le k^{eme} chemin local de la classe C_i qui permet d'aller de a à b . Supposons que le nombre de chemins locaux possibles entre a et b relatifs à la classe C_i est $n_{p_i(a,b)}$. Le plus court chemin local associé à la classe C_i , et qui permet d'aller de a vers b est un chemin $p_i^*(a, b)$ avec :

$$d(p_i^*(a, b)) = \min_{k=1,2,\dots,n_{p_i(a,b)}} \{d(p_i^k(a, b))\}$$

On note $p^k(A, B)$ le k^{eme} chemin global du graphe G qui permet d'aller de A à B . Supposons que le nombre de chemins globaux possibles entre A et B est $n_{p(A,B)}$. Le plus court chemin global, et qui permet d'aller de A vers B est un chemin $p^*(A, B)$ avec :

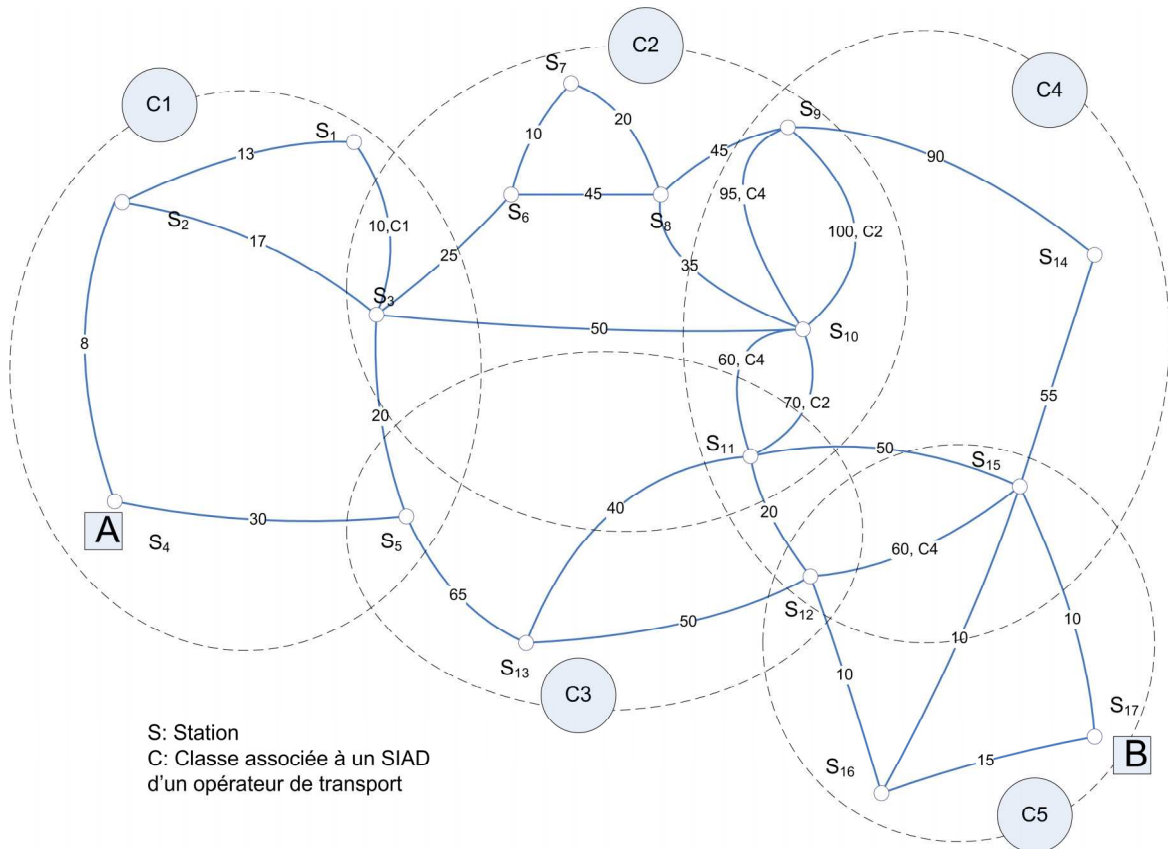
$$d(p^*(A, B)) = \min_{k=1,2,\dots,n_{p(A,B)}} \{d(p^k(A, B))\}$$

Exemple Cette modélisation peut être illustrée par la figure 3.2. $G(N, E)$ est dans ce cas un graphe composé de cinq classes $\{C_1, C_2, C_3, C_4, C_5\}$ avec les ensembles de noeuds suivants : $N_1 = \{s_1, s_2, s_3, s_4, s_5\}$, $N_2 = \{s_3, s_6, s_7, s_8, s_9, s_{10}, s_{11}\}$, $N_3 = \{s_5, s_{11}, s_{12}, s_{13}\}$, $N_4 = \{s_9, s_{10}, s_{11}, s_{12}, s_{14}, s_{15}\}$; et $N_5 = \{s_{12}, s_{15}, s_{16}, s_{17}\}$.

les ensembles d'intersection $N_1 \cap N_2 = \{s_3\}$, $N_1 \cap N_3 = \{s_5\}$, $N_2 \cap N_3 = \{s_{11}\}$, $N_2 \cap N_4 = \{s_9, s_{10}, s_{11}\}$, $N_3 \cap N_4 = \{s_{11}, s_{12}\}$, $N_3 \cap N_5 = \{s_{12}\}$ et $N_4 \cap N_5 = \{s_{12}, s_{15}\}$ définissent les ensembles de pôles d'échanges qui permettent, au cours d'un déplacement, de changer de réseau de transport.

Un arc dans ce graphe définit une course assurée par un seul et unique moyen de transport géré par un seul opérateur. Cependant, on notera (d, C_i) le poids $d(s_1, s_2)_i$ de l'arc $(s_1, s_2)_i$

Pour aller de s_3 à s_{11} , on peut emprunter un itinéraire local de la classe C_2 tel que $\{s_3 - s_{10} - s_{11}\}$. Ce dernier est qualifié d'intra-classe puisqu'il est composé des arcs $\{(s_3, s_{10}), (s_{10}, s_{11})_2\}$. On peut aussi emprunter un itinéraire global et interclasse tel que $\{s_3 - s_{10} - s_{11}\}$ composé des arcs $\{(s_3, s_{10}), (s_{10}, s_{11})_4\}$.

FIG. 3.2 – Exemple d'un graphe $G(N, E)$ dans un environnement distribué

Si $G(N, E)$ était centralisé ou connu à l'avance, pour calculer le plus court chemin $P_G^*(A, B)$ sur ce graphe $G(N, E)$, d'une origine A correspondant à s_4 vers une origine B correspondant à s_{17} , il suffirait d'exécuter un algorithme de plus court chemin tel que Dijkstra sur ce graphe. Dans ce cas le plus court chemin sera $P_G^*(A, B) = \{s_4 - s_5 - s_{13} - s_{12} - s_{16} - s_{17}\}$.

3.3.2 Principe du plus court chemin distribué

Dans la littérature, rares sont les travaux et les algorithmes qui se sont intéressés à cette problématique. Wang et Kaempke [Wang et al.04] présente le premier algorithme de plus court chemin dans un environnement distribué. Cette nouvelle approche se base sur la construction d'un graphe d'intersection complet, noté G_{cint} , qui ne considère que les noeuds d'intersections entre les classes C_i , et les meilleurs itinéraires locaux qui permettent de les relier. L'algorithme Dijkstra sera par la suite exécuté sur le G_{cint} . Cet algorithme distribué est de complexité $O(n^2)$ où n est le nombre de noeuds d'intersection.

3.3.2.1 Le graphe d'intersection complet

[Wang et al.04] définissent le graphe complet d'intersection $G_{cint}(N_{cint}, E_{cint})$ correspondant au graphe distribué $G(N, E)$ comme suit :

- les noeuds de G_{cint} sont les noeuds d'intersection des n classes de $G(N, E)$:

$$N_{cint} = \bigcup_{i=1, j=1}^n (N_i \cap N_j)$$

- (a, b) est un arc de E_{cint} , si et seulement si $a \in N_{cint}$ et $b \in N_{cint}$ et $\{a, b\} \subset N_i$ ensemble de noeuds d'une classe C_i . Dans ce cas $d(a, b) = d(p_i^*(a, b))$. En d'autres termes, on associe à l'arc (a, b) la meilleure offre locale qui peut être fournie par la classe C_i pour aller de a vers b . Dans le cas général où plusieurs classes contiennent le couple $\{a, b\}$, on note $I(a, b)$ l'ensemble des index de ces classes C_i . Ainsi $I(a, b) = \{j | \{a, b\} \in C_j\}$. Le poids associé à l'arc (a, b) sera la meilleure offre parmi l'ensemble des meilleures offres locales des différents C_i :

$$d(a, b) = \min_{i \in I(a, b)} d(p_i^*(a, b))$$

Une deuxième étiquette r , sera associée à l'arc (a, b) pour marquer à quelle classe C_i , il est associé ; et par conséquent à quel réseau de transport appartient le tronçon d'itinéraire qui a été retenu. On note :

$$r(a, b) = C_i \quad \text{si} \quad d(a, b) = d(p_i^*(a, b)) \quad \text{avec} \quad i \in I(a, b)$$

A titre d'exemple si $d(a, b) = d(p_j^*(a, b))$, alors l'arc (a, b) n'utilisera que des offres locales de la classe C_j . Finalement, chaque arc (a, b) de G_{cint} sera associé un couple d'étiquettes (d, r) ; d étant la durée et r la classe considérée C_j .

En appliquant la définition précédente du graphe d'intersection complet associé au graphe $G(N, E)$ de la figure 3.2 nous construisons le G_{cint} de la figure 3.3.

On commence par déterminer les noeuds d'intersection. Dans ce cas, l'ensemble de ces noeuds est $N_{cint} = \{s_3, s_5, s_9, s_{10}, s_{11}, s_{12}, s_{15}\}$.

Pour chaque classe C_i , on détermine ensuite l'ensemble $N_{cint i}$ des noeuds d'intersection qu'elle partage avec les autres classes. Ainsi $N_{cint 1} = \{s_3, s_5\}$, $N_{cint 2} = \{s_3, s_9, s_{10}, s_{11}\}$, $N_{cint 3} = \{s_5, s_{11}, s_{12}\}$, $N_{cint 4} = \{s_9, s_{10}, s_{11}, s_{12}, s_{15}\}$, $N_{cint 5} = \{s_{12}, s_{15}\}$.

C_i	arc (a, b)	$d(p_i^*(a, b))$	$p_i^*(a, b)$	$\min_{i \in I(a,b)} d(p_i^*(a, b))$	(d, r)
C_1	(s_3, s_5)	20	$\{s_3 - s_5\}$	X	$(20, C_1)$
C_2	(s_3, s_9)	100	$\{s_3 - s_6 - s_7 - s_8 - s_9 - s_{10}\}$	X	$(100, C_2)$
	$(s_{10}, s_{11})_2$	70	$\{s_{10} - s_{11}\}$		
	$(s_9, s_{10})_2$	100	$\{s_9 - s_{10}\}$		
	(s_3, s_{11})	120	$\{s_3 - s_{10} - s_{11}\}$	X	$(120, C_2)$
	$(s_9, s_{11})_2$	140	$\{s_9 - s_8 - s_{10} - s_{11}\}$	X	$(140, C_2)$
	(s_3, s_{10})	50	$\{s_3 - s_{10}\}$	X	$(50, C_2)$
C_3	(s_5, s_{11})	105	$\{s_5 - s_{13} - s_{11}\}$	X	$(105, C_3)$
	$(s_{11}, s_{12})_3$	20	$\{s_{11} - s_{12}\}$	X	$(20, C_3)$
	(s_5, s_{12})	115	$\{s_5 - s_{13} - s_{12}\}$	X	$(115, C_3)$
C_4	$(s_9, s_{10})_4$	95	$\{s_9 - s_{10}\}$	X	$(95, C_4)$
	$(s_{10}, s_{11})_4$	60	$\{s_{10} - s_{11}\}$	X	$(60, C_4)$
	$(s_{11}, s_{12})_4$	20	$\{s_{11} - s_{12}\}$		
	$(s_{12}, s_{15})_4$	60	$\{s_{12} - s_{15}\}$		
	(s_9, s_{15})	135	$\{s_9 - s_{14} - s_{15}\}$	X	$(135, C_4)$
	$(s_9, s_{11})_4$	155	$\{s_9 - s_{10} - s_{11}\}$		
	(s_{10}, s_{12})	80	$\{s_{10} - s_{11} - s_{12}\}$	X	$(80, C_4)$
	(s_{10}, s_{15})	110	$\{s_{12} - s_{11} - s_{15}\}$	X	$(110, C_4)$
	(s_{11}, s_{15})	50	$\{s_{11} - s_{15}\}$	X	$(50, C_4)$
	(s_9, s_{12})	175	$\{s_9 - s_{10} - s_{11} - s_{12}\}$	X	$(175, C_4)$
C_5	$(s_{12}, s_{15})_5$	20	$\{s_{12} - s_{16} - s_{15}\}$	X	$(20, C_5)$

TAB. 3.1 – Construction du G_{cint}

Pour chaque classe C_i , on détermine l'ensemble des chemins $p_i^*(a, b)$, qui permettent de relier de différentes manières possibles des noeuds d'intersection de la même classe ($a \in N_{cint\ i}$ et $b \in N_{cint\ i}$). Ces chemins $p_i^*(a, b)$ constitueront des éléments candidats à l'ensemble E_{cint} des arcs (a,b) associés à la classe C_i sur le graphe G_{cint} . Pour une classe C_i qui partage n_i noeuds d'intersection avec les autres classes, le nombre de chemin candidats à déterminer, et par conséquent le nombre des éléments candidats à $E_{cint\ i}$ sera $np_i^* = \frac{n_i(n_i-1)}{2}$. On détermine ensuite $d(p_i^*(a, b))$ pour chaque arc candidat.

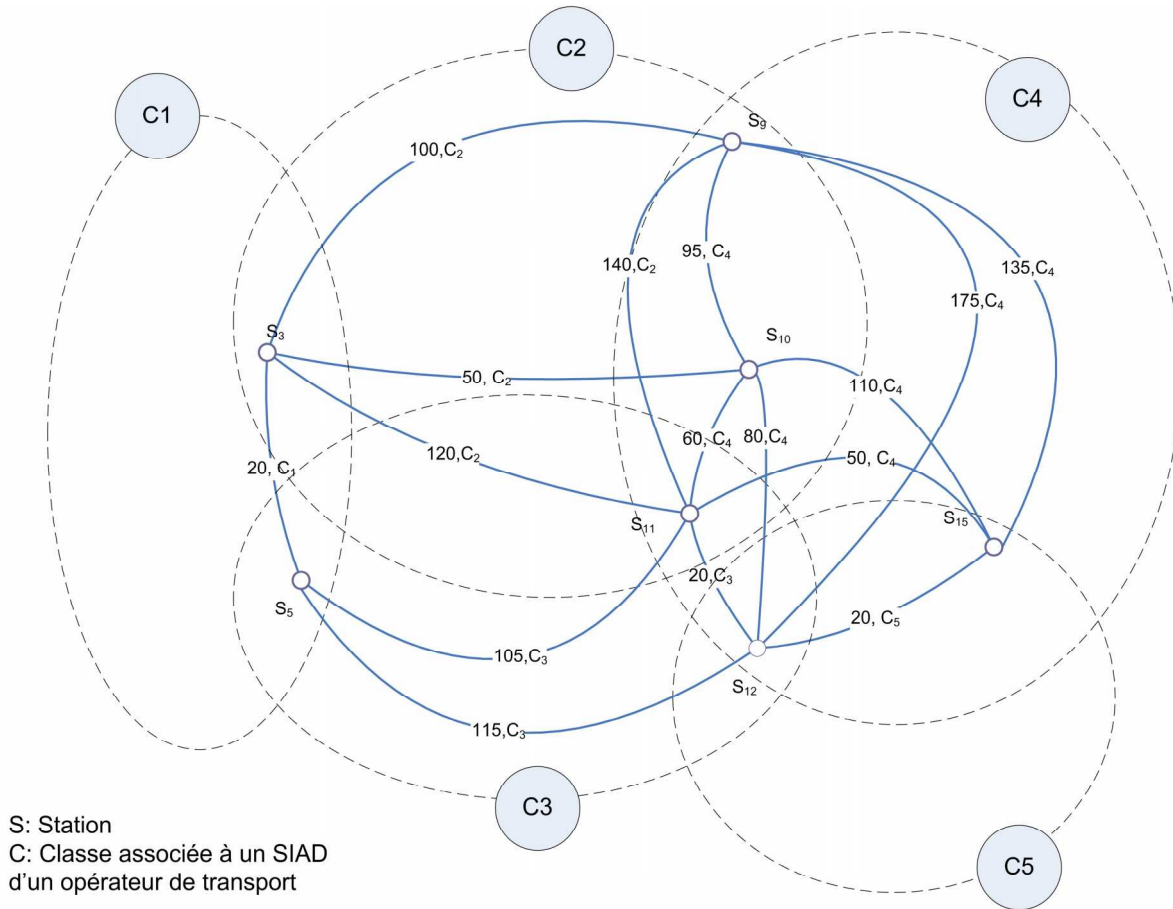


FIG. 3.3 – Exemple d'un graphe $G_{cint}(N_{cint}, E_{cint})$ dans un environnement distribué

A titre d'exemple, la classe C_2 partage $n_2 = 4$ noeuds d'intersection avec les autres classes d'où, le nombre d'arcs candidats à déterminer est $np_2^* = \frac{n_2(n_2-1)}{2} = 6$. Il s'agit en effet des chemins candidats $p_2^*(s_3, s_9)$, $p_2^*(s_{10}, s_{11})$, $p_2^*(s_9, s_{10})$, $p_2^*(s_3, s_{11})$, $p_2^*(s_9, s_{11})$, $p_2^*(s_3, s_{10})$.

Une fois les offres locales déterminées pour chaque classe C_i , l'étape suivante consiste à déterminer parmi les arcs candidats, les arcs qui seront retenus pour le graphe d'intersection

complet G_{cint} . Ces arcs (a, b) vérifient la condition $d(a, b) = \min_{i \in I(a, b)} d(p_i^*(a, b))$ avec $i \in I(a, b)$; $I(a, b)$ défini comme l'ensemble des index des classes qui présentent des offres locales pour un arc (a, b) .

En d'autres termes si plusieurs chemins candidats ont été sélectionnés pour un couple $\{a, b\}$, le chemin retenu pour l'arc (a, b) du graphe G_{cint} , sera le chemin qui présente la meilleure offre et par conséquent le poids minimal. Dans notre cas (voir le tableau Tab 3.1), pour l'arc (s_9, s_{11}) , C_2 présente un chemin candidat $(s_9, s_{11})_2 = \{s_9 - s_8 - s_{10} - s_{11}\}$ de durée 140. C_4 présente aussi une offre locale $(s_9, s_{11})_4 = \{s_9 - s_{10} - s_{11}\}$ de durée 155. Dans ce cas, $I(s_9, s_{11}) = \{2, 4\}$. L'offre retenue pour former l'arc (s_9, s_{11}) de G_{cint} sera celle de C_2 (voir le tableau Tab 3.1). L'arc (s_9, s_{11}) est labellisé $(140, C_2)$ sur le graphe d'intersection complet.

D'une manière générale l'arc (a, b) de G_{cint} sera le chemin p_i^k entre deux stations d'intersection a et b , relatif à une classe C_i et qui vérifie :

$$d(a, b) = \min_{i \in I(a, b)} (\min_{k=1, 2, \dots, n_{p_i(a, b)}} (\{d(p_i^k(a, b))\}))$$

3.3.2.2 Le graphe d'intersection étendu

Pour calculer le plus court chemin global entre deux noeuds A et B du graphe G , [Wang et al.04] définissent aussi un graphe étendu d'intersection.

On notera ce graphe $G_{eint(A, B)}(N_{eint(A, B)}, E_{eint(A, B)})$. Il est défini comme suit :

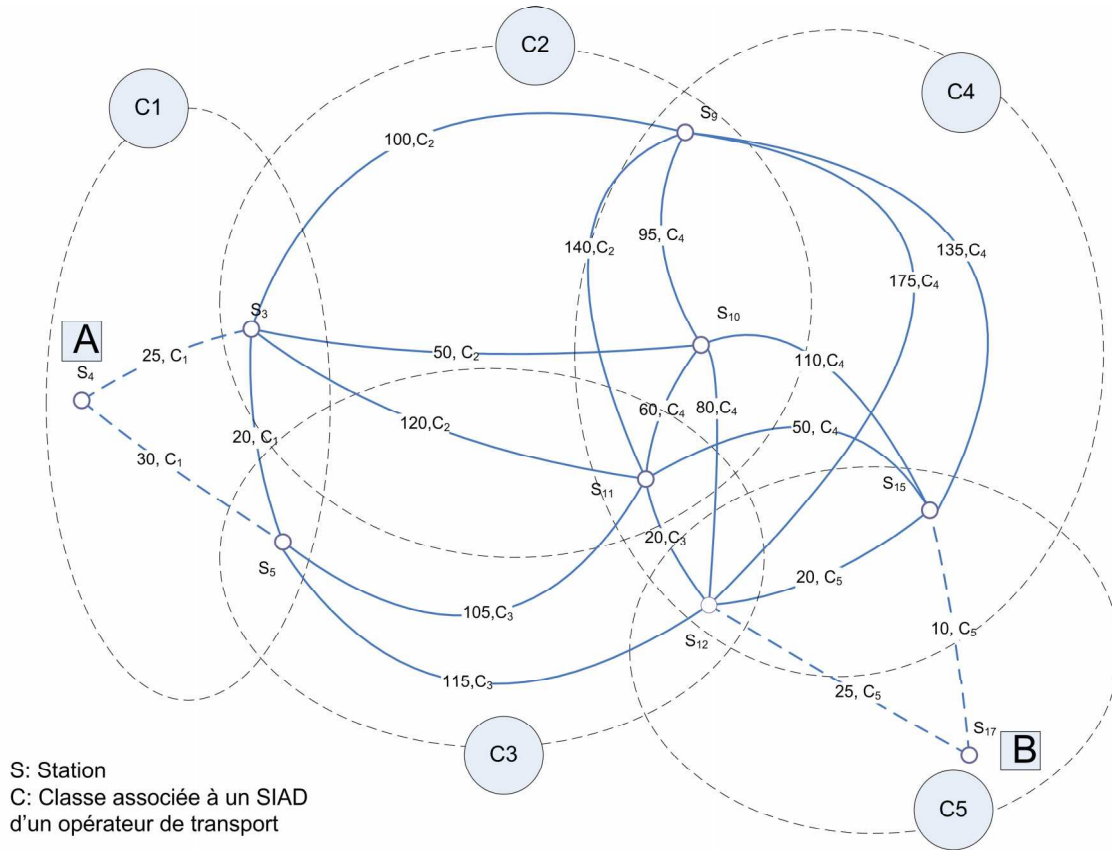
- les noeuds de $G_{eint(A, B)}$ sont tous les noeuds de G_{cint} en plus des noeuds A et B , s'ils ne sont pas déjà inclus dans G_{cint} . Cet ensemble est défini par :

$$N_{eint(A, B)} = N_{cint} \cup \{A, B\}$$

- les arcs de $G_{eint(A, B)}$ sont tous les arcs de G_{cint} aux quels on ajoute, l'ensemble des arcs relatifs aux plus courts chemin locaux qui permettent de relier A et B , aux noeuds d'intersection des classes relatives à ces noeuds de départ et d'arrivée. Soit i_A l'index de la classe qui contient le noeud A , et i_B l'index de la classe qui contient le noeud B . E_{cint}^A est l'ensemble des arcs (A, a) relatifs aux chemins locaux $p_{i_A}^*(A, a)$ de C_{i_A} et qui vérifient :

$$d(A, a) = \min_{k=1, 2, \dots, n_{p_{i_A}(A, a)}} (\{d(p_{i_A}^k(A, a))\})$$

De même, E_B^{cint} est l'ensemble des arcs (b, B) relatifs aux chemins locaux $p_{i_B}^*(b, B)$ de C_{i_B} et qui vérifient :

FIG. 3.4 – Exemple d'un graphe d'intersection étendu $G_{eint}(N_{eint}, E_{eint})$

$$d(b, B) = \min_{k=1,2,\dots,n_{p_i(b,B)}} (\{d(p_{i_B}^k(b, B))\})$$

l'ensemble des arcs E_{eint} est finalement défini par :

$$E_{eint}(A,B) = E_{eint}^A \cup E_{eint} \cup E_B^{eint}$$

Dans l'exemple précédent on associe s_4 au noeud de départ A , et s_{17} au noeud d'arrivée B . Nous constatons que $s_4 \in C_1$ et $s_{17} \in C_5$. Les noeuds d'intersection de C_1 qui permettent d'accéder depuis C_1 au graphe G_{eint} sont $N_{1\ eint} = \{s_3, s_5\}$. On calcule alors $p_1^*(s_4, s_3) = \{s_4 - s_2 - s_3\}$ avec $d(p_1^*(s_4, s_3)) = 25$. De même, $p_1^*(s_4, s_5) = \{s_4 - s_5\}$ avec $d(p_1^*(s_4, s_5)) = 30$. Ces chemins constitueront les arcs (s_4, s_3) (s_4, s_5) , les éléments de $E_{eint}^{s_4}$.

Par ailleurs, nous calculons aussi les chemins optimaux locaux relatifs à C_5 et qui permettent d'accéder depuis les noeuds d'intersections définis dans $N_{5\ eint} = \{s_{12}, s_{15}\}$ au noeud

d'arrivée s_{17} . Ces chemins sont $p_5^*(s_{12}, s_{17}) = \{s_{12} - s_{26} - s_{17}\}$ avec $d(p_5^*(s_{12}, s_{17})) = 25$ et $p_5^*(s_{15}, s_{17}) = \{s_{15} - s_{17}\}$ avec $d(p_5^*(s_{15}, s_{17})) = 10$. Ces arcs (s_{12}, s_{17}) (s_{15}, s_{17}) , constituent les éléments de $E_{s_{17}}^{cint}$ (voir Fig 3.4).

3.3.2.3 L'Algorithme

L'algorithme de plus court chemin distribué (PCCD) qui permet de trouver un plus court chemin $P_G^*(A, B)$ d'un noeud A vers un noeud B sur le graphe $G(N, E)$, consiste à construire un graphe d'intersection étendu $G_{eint(A,B)}$ associé à $G(N, E)$ et d'appliquer ensuite sur ce dernier un plus court chemin classique tel que Dijkstra.

l'algorithme (PCCD) est résumé par les étapes suivantes (Fig 3.5) :

1. Construire le G_{eint} associé au $G(N, E)$.
2. Construire le $G_{eint}(A, B)$ associé au $G(N, E)$.
3. Déterminer $P_{G_{eint}(A,B)}^* = \{s_1 - \dots - s_m\}$ en exécutant Dijkstra sur $G_{eint}(A, B)$.
 $s_1 - \dots - s_m$ sont les noeuds d'intersection éléments de $P_{G_{eint}(A,B)}^*$
4. Pour $k = 1$ à $k = m - 1$
 - (a) pour (s_k, s_{k+1}) déterminer la classe C_i relative à l'arc (s_k, s_{k+1})
 - (b) pour (s_k, s_{k+1}) déterminer le détail de $p_i^*(s_k, s_{k+1})$
5. les $p_i^*(s_k, s_{k+1})$ constituent le chemin global optimal $P_G^*(A, B)$

FIG. 3.5 – Algorithme de plus court chemin distribué

Pour démontrer l'optimalité de cet algorithme, [Wang et al.04] proposent et démontrent les deux lemmes suivants :

Lemme 1 *Un chemin $P_{G_{eint}(A,B)}$ de $G_{eint}(A,B)$ de longueur globale $d(P_{G_{eint}(A,B)}) = L$ (avec $L < \infty$) correspond à un chemin $P_G(A, B)$ sur $G(N, E)$ avec la même longueur $d(P_G(A, B)) = L$.*

Lemme 2 *S'il existe un plus court chemin $P_G^*(A, B)$ sur $G(N, E)$ alors il existe un chemin $P_{G_{eint}(A,B)}^*(A, B)$, qui contient les mêmes noeuds d'intersection contenus dans $P_G^*(A, B)$, et tel que :*

$$d(P_{G_{eint}(A,B)}^*(A, B)) = d(P_G^*(A, B))$$

Preuve Le Lemme 2 peut être démontré facilement. En effet Si $P_G^*(A, B)$ est le plus court chemin sur $G(N, E)$, alors il passe obligatoirement de A vers B par des noeuds d'intersection ($P_G^*(A, B) = \{A = s_1 - ..s_k - s_m = B\}$). Et puisque le plus court chemin est toujours composé de plus courts chemins, alors $P_G^*(A, B)$ est composé des meilleures offres locales, $p^*(A, s_2)$, $p^*(s_2, s_3) \dots p^*(s_{m-1}, b)$. Or, ces éléments font partie de $G_{eint(A,B)}$, par définition.

Donc il existe un chemin $P_{G_{eint(A,B)}}^*(A, B)$ qui correspond à $P_G^*(A, B)$ qui passe par les mêmes noeuds d'intersection et de même longueur.

En se basant sur ces deux lemmes, [Wang et al.04] démontrent le théorème suivant :

théorème 1 *l'algorithme PCCD calcule le plus court chemin $P_G^*(A, B)$ sur le graphe $G(N, E)$.*

Preuve soit $P_{G_{eint(A,B)}}^*(A, B)$ le plus court chemin de A à B sur le graphe $G_{eint(A,B)}$ calculé à l'étape 2 du PCCD. D'après le Lemme 1, ce chemin correspond à un chemin $P_G^*(A, B)$ sur le graphe $G(N, E)$ avec $d(P_{G_{eint(A,B)}}^*(A, B)) = d(P_G^*(A, B))$.

S'il existe un autre chemin sur $G(N, E)$ qui est plus court que $P_G^*(A, B)$, alors d'après le Lemme 2, il doit exister un chemin qui lui correspond sur $G_{eint(A,B)}$. Ceci est en contradiction avec le fait que $P_{G_{eint(A,B)}}^*(A, B)$ est le plus court chemin de A à B sur le graphe $G_{eint(A,B)}$.

Par conséquent $P_G^*(A, B)$ obtenu à l'étape 3, par rapport à $P_{G_{eint(A,B)}}^*(A, B)$, est le plus court chemin de A vers B sur le graphe $G(N, E)$.

Nous pouvons aussi démontrer ceci autrement :

Soit $P_G^*(A, B)$ de longueur L , le plus court chemin sur $G(N, E)$. D'après le Lemme 2, il lui correspond un chemin $P_{G_{eint(A,B)}}^*(A, B)$ de même longueur sur $G_{eint(A,B)}$.

Supposons ensuite, que $P_{G_{eint(A,B)}}^{*'}(A, B)$ est le plus court chemin calculé lors de l'étape 3. Ce chemin est de durée L' tel que $L' < L$. D'après le Lemme 1, il lui correspond un chemin $P_G^{*'}(A, B)$ de longueur $L' < L$.

Hors ceci est en contradiction avec le fait que $P_G^*(A, B)$ est le plus court chemin sur $G(N, E)$. Par conséquent $P_G^*(A, B)$ obtenu à l'étape 3, par rapport à $P_{G_{eint(A,B)}}^*(A, B)$, est le plus court chemin de A vers B sur le graphe $G(N, E)$.

Exemple Si nous appliquons cet algorithme sur notre exemple $G(N, E)$ de la figure 3.2, en appliquant le plus court chemin sur le graphe $G_{eint(s_4, s_{17})}$ de la figure 3.4, nous obtenons le chemin composé des noeuds d'intersection suivants :

$$P_{G_{eint}(s_4, s_{17})}^*(s_4, s_{17}) = \{s_4 - s_5 - s_{12} - s_{17}\}$$

Ce qui correspond au chemin détaillé suivant (voir Tab 3.1) :

$$P_G^*(s_4, s_{17}) = \{s_4 - s_5 - s_{13} - s_{12} - s_{16} - s_{17}\} \quad avec \quad d(P_G^*(s_4, s_{17})) = 170$$

3.3.2.4 PCCD et Organisation SMA

Dans le SICM, nous associons un agent d'information (AI) à chaque SIAD d'un opérateur de transport. Chaque AI correspond alors à une classe. L'AI_{*i*} est capable de répondre à une requête d'itinéraire $req(a, b)$ par un plus court chemin $p_i^*(a, b)$ qui permet d'aller de a vers b sur un réseau de transport modélisé par un graphe G_i .

Pour répondre à une requête globale $Req(A, B)$ sur le graphe global et distribué $G(N, E)$ L'algorithme PCCD sera exécuté par un agent médiateur. Il s'agit de l'agent courtier (AC) compositeur d'itinéraire, défini précédemment (§2.4.2.2).

L'agent médiateur ou courtier commence par construire le graphe d'intersection complet G_{cint} et le graphe d'intersection étendu $G_{eint(A, B)}$ correspondant à la requête $req(A, B)$, en demandant les meilleures offres locales aux différents agents d'information impliqués. Le médiateur exécute ensuite l'algorithme Dijkstra sur ce dernier pour déterminer la meilleure offre globale qui répond à la requête $Req(A, B)$. Une fois la meilleure offre globale déterminée, cet agent va demander le détail de l'offre locale retenue : l'ensemble des stations locales par lesquelles passe l'itinéraire présenté par cette offre. La composition et la recherche de l'itinéraire est assurée par l'algorithme PCCD : compétence nécessaire de l'agent AC.

Dans cette composition, on suppose que l'agent AC a une vision globale de tous les AI de l'environnement. Le domaine de recherche n'est pas limité ; il est associé à l'ensemble des AI qui définissent l'environnement. Dans ce cas, l'organisation Interface-Mediateur-Adaptateur est celle qui s'adapte le mieux à l'utilisation d'un tel algorithme, puisque le médiateur n'a pas besoin de limiter son domaine de recherche. On associe ainsi à chaque utilisateur un médiateur.

Par ailleurs il n'est pas nécessaire dans ce cas, de reconstruire à chaque requête de l'utilisateur, le graphe G_{cint} , puisque ce dernier est indépendant des requêtes du type $Req(A, B)$. Par contre, pour construire le graphe $G_{eint(A, B)}$, il est nécessaire d'avoir le couple de noeuds

$\{A, B\}$, pour pouvoir ensuite récupérer les chemins $p_{i_A}^*(A, a)$ et $p_{i_B}^*(b, B)$ qui relient A et B aux noeuds d'intersections a et b du graphe d'intersection complet. En, définitif, en utilisant cet algorithme, puisque l'ensemble des AI , ne change pas, et le G_{cint} reste aussi le même, il suffit de reconstruire le $G_{cint(A,B)}$ pour répondre à une nouvelle requête utilisateur $Req(A, B)$.

A ce stade on a construit le $G_{cint(A,B)}$, et on exécute le PCCD pour chercher et composer une offre globale qui minimise un seul critère. Ce critère peut être la durée des trajets, le coût des billets de transport ou le nombre de correspondances.

Néanmoins cette solution reste envisageable seulement dans le cas où les poids des arcs sont constants. Mais si ces poids varient en fonction, par exemple, des temps de départs possibles, des heures de pointes ou d'autres paramètres du contexte, le modèle adopté et l'algorithme PCCD retenu deviennent inadaptés. Ceci est généralement le cas dans le transport public.

3.4 Algorithme de plus court chemin distribué et horaire

3.4.1 Modélisation du problème

Dans le domaine des transports publics, les réseaux sont dits dynamiques et dépendants du temps. En effet entre une station de départ a et une station d'arrivée b , on peut avoir plusieurs courses (transferts programmés), et par conséquent plusieurs heures de départs possibles de la station a . A la différence des réseaux de transports statiques, la durée du trajet, ou même le coût, devient dans ce cas une fonction du temps de départ à la station a . On s'intéressera dans la suite au seul critère durée du voyage entre a et b , qui est généralement le premier critère à considérer dans les transports publics.

Nous nous proposons dans la suite d'explorer les algorithmes qui permettent de calculer le plus court chemin sur un graphe dynamique d'une station a vers une station b . Il s'agit dans ce cas de trouver le chemin qui permet de minimiser le temps global de voyage entre a et b . Ce temps comprend non seulement le temps de trajet que passerait un voyageur dans les moyens de transport, mais aussi le temps d'attente dans les gares et les stations, que passerait ce dernier à attendre un prochain départ dans une station.

Cette exploration nous permettra d'adapter les algorithmes existants à la problématique

de recherche et de composition d'itinéraires horaires dans un environnement distribué composé de différents réseaux de transports dynamiques.

Pour ce faire, nous commençons par définir la notation suivante :

Soit $G(N, E)$ un graphe orienté. G est dit dynamique si la durée de l'arc (i, j) est une fonction $d_{ij}(t)$ du temps t de départ de la station i . Pour la suite on suppose que $d_{ij}(t) > 0$.

On note $N^+(i)$ l'ensemble des noeuds adjacents et successeurs, sur le graphe G , du noeud i . $N^+(i)$ est l'ensemble $\{j | (i, j) \in E\}$. De même, on définit $N^-(i)$ comme étant l'ensemble des noeuds adjacents et prédécesseurs du noeud i sur le graphe G . $N^-(i)$ est l'ensemble $\{j | (j, i) \in E\}$.

Nous présentons dans la suite deux approches utilisées dans la littérature pour modéliser les graphes dynamiques dans le transport. Il s'agit du modèle dépendant du temps ou "time dependent model" et du modèle espace temps appelé aussi "time space model" ou "time-expanded model". Pour chaque modèle, des algorithmes ont été développés pour résoudre le problème du plus court chemin sur un graphe dynamique.

3.4.1.1 Modèle dépendant du temps d'un réseau dynamique

Le modèle dépendant du temps modélise le réseau dynamique par le graphe $G(N, E)$ défini précédemment. En effet nous associons, pour chaque arc (i, j) une liste de départs possibles planifiés par l'opérateur qu'on notera $TD(i, j) = (t_{d_1}(i, j), t_{d_2}(i, j), \dots, t_{d_k}(i, j), \dots, t_{d_{L(i,j)}}(i, j))$, avec $t_{d_k}(i, j)$ le k ème départ possible associé à l'arc (i, j) et $L(i, j)$ le nombre total de ces départs.

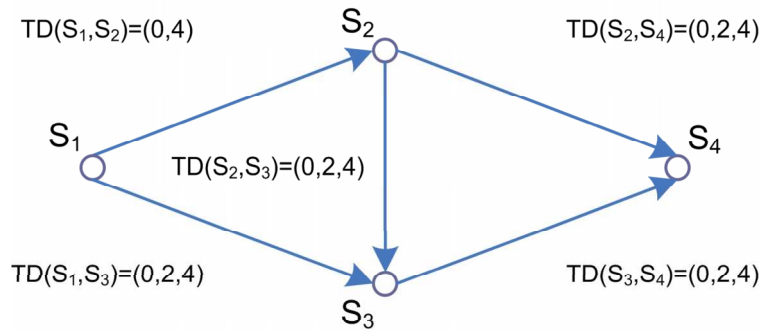
Chaque station du réseau est modélisée par un seul noeud. Et à chaque temps de départ possible $t_{d_k}(i, j)$, et par conséquent à chaque course qui permet d'aller de i vers j , on associe une durée $d_{ij}(t_{d_k}(i, j))$. La durée de l'arc (i, j) est alors définie par une fonction du temps d'arrivée t_i à la station de départ i . Ce temps t_i est défini comme le temps de départ souhaité. $d_{ij}(t_i)$ est définie comme suit :

$$d_{ij}(t_i) = \begin{cases} d_{ij}(t_{d_k}(i, j)) & \text{si } t_i = t_{d_k}(i, j) \\ d_{ij}(t_{d_{k+1}}(i, j)) & \text{si } t_{d_k}(i, j) < t_i < t_{d_{k+1}}(i, j) \end{cases}$$

$d_{ij}(t_i)$ dépend, en effet, du temps de départ effectif de la course. Ainsi l'utilisateur peut attendre dans la station i de départ. On montre dans la suite que des approches de ré-

solution se sont basées sur ce modèle pour répondre au problème du temps de voyage minimum[Chabini et al.98],[Schulz05]. En pratique, ce modèle revient à mémoriser pour chaque arc (i, j) , l'ensemble des départs possibles $TD(i, j)$; et pour chaque temps de départ possible $t_{d_k}(i, j)$, la durée $d_{ij}(t_{d_k}(i, j))$ qui lui correspond. L'ensemble des durées $d_{ij}(t_{d_k}(i, j))$ est mémorisé dans une table.

La figure (Fig 3.6) représente un graphe dynamique modélisé par un modèle dépendant du temps.



		$d_{ij}(t_{dk}(S_i, S_j))$				
$TD(S_i, S_j)$		$d_{12}(t)$	$d_{13}(t)$	$d_{23}(t)$	$d_{24}(t)$	$d_{34}(t)$
0		1	2	1	1	1
2			1	1	2	1
4		1	1	1	1	1

FIG. 3.6 – Exemple d'un modèle dépendant du temps

3.4.1.2 Modèle espace temps d'un réseau dynamique.

Pour résoudre ce problème de temps de voyage minimum, d'autres travaux adoptent une transformation du graphe $G(N, E)$. Il s'agit en effet d'associer à ce dernier un graphe $G^*(N^*, E^*)$ dit modèle espace-temps de G . Ainsi à chaque temps de départ t d'un noeud i de G , on associe un couple (i, t) qui représente un noeud sur G^* . N^* est alors défini comme l'ensemble des noeuds $N^* = \{(i, t) | i \in N, t \in \{t_{d_1}(i, j), t_{d_2}(i, j), \dots, t_{d_k}(i, j), \dots, t_{d_L(i, j)}(i, j)\}\}$ et ce pour tout $j \in N^+(i)$. E^* est l'ensemble des arcs qui vont d'un noeud (i, t) vers un noeud

$(j, t + d_{ij}(t))$ avec $j \in N^+(i)$. Si par ailleurs, le temps d'arrivée $t + d_{ij}(t)$ ne correspond pas à un temps de départ sur j , on crée alors des noeuds du type $(i, t + d_{ij}(t))$ qui correspondent à l'évènement d'arrivée d'un noeud vers un autre. De plus, si l'attente est autorisée dans les noeuds de G , on peut créer dans G^* des arcs du type $((i, t_{d_k}(i, j)), (i, t_{d_{k+1}}(i, j)))$.

Ainsi dans le graphe G^* , chaque noeud de G sera dupliqué en un ensemble de noeuds, qui correspondent aux évènements de départs et d'arrivées relatifs à ce noeud. Les arcs de G^* , modélisent alors les transferts entre stations ou les attentes dans la même station.

Le modèle espace-temps permet de transformer un graphe dynamique G en un graphe statique G^* où la durée des arcs n'est plus fonction du temps. De ce fait résoudre un problème de temps de voyage minimum, sur le graphe G revient à résoudre un problème de plus court chemin classique sur le graphe G^* [Chabini et al.98], [Pallottino et al.97], [Schulz05].

La figure suivante (Fig 3.7) reprend le graphe G de l'exemple précédent (Fig 3.6), transformé en un graphe espace-temps G^* .

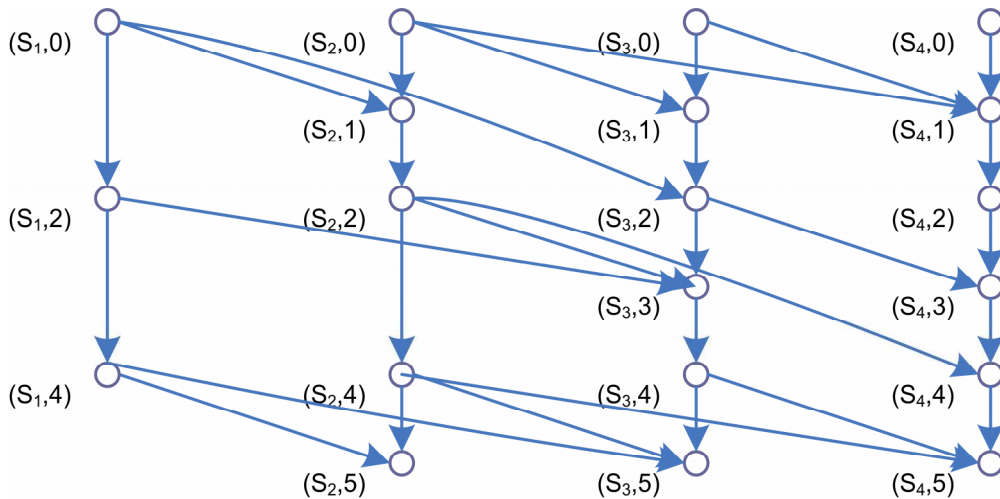


FIG. 3.7 – Exemple d'un modèle espace-temps

3.4.1.3 Réseaux dynamiques FIFO et non-FIFO

Pour résoudre le problème de temps de voyage minimum d'autres travaux, et notamment ceux de [Chabini et al.98] se basent sur les graphes dynamiques modélisés par le modèle dépendant du temps. En effet, au lieu de transformer leurs modèles, ces derniers adaptent les algorithmes de plus court chemin tel que Dijkstra pour qu'ils prennent en considération cet aspect dynamique : plusieurs départs possibles d'un noeud i vers un noeud j , avec des durées

de trajets $d_{ij}(t)$ qui sont des fonctions du temps de départ t de la station i . Ce problème de temps de voyage minimum, est souvent associé à deux variantes :

- Un temps de voyage minimum avec une heure de départ au plus tôt de la station de départ. Il s'agit dans ce cas d'un problème de temps d'arrivée au plus tôt à la station d'arrivée.
- Un temps de voyage minimum avec une heure d'arrivée au plus tard à la station d'arrivée. Il s'agit dans ce cas d'un problème de temps de départ au plus tard de la station de départ.

Nous considérons dans la suite le problème de minimisation de temps d'arrivée à une station j , en partant d'une station i à un instant t . les algorithmes présentés par [Chabini et al.98] pour résoudre ce problème seront détaillés dans la suite.

Un réseau de transport est dit FIFO (First In First Out) si et seulement si pour tout arc (i, j) du graphe G , on vérifie la propriété FIFO suivante :

$$t + d_{ij}(t) \leq t' + d_{ij}(t') \quad \text{pour } \forall \{t, t' | t \leq t'\}$$

En d'autres termes, si la propriété FIFO est vérifiée par tous les arcs de G , il ne peut exister -à la différence des réseaux non-FIFO - deux trajets de i vers j , tel que le premier partant de i , arrive en deuxième à j . En effet dans les réseaux non-FIFO, à une station donnée, il s'avère parfois nécessaire de laisser un bus ou un métro passer et de prendre le prochain afin d'arriver au plus tôt à destination.

Cette différence peut être illustrée par la figure (Fig 3.8).

Les lemmes suivants, démontrés par [Chabini et al.98], présentent des caractéristiques intrinsèques des chemins associés aux réseaux FIFO. En effet :

Lemme 3 *Dans un réseau FIFO, si $p(i, j, t)$ est un chemin qui part de i à l'instant t vers un noeud j , alors $p(i, j, t)$ vérifie la propriété FIFO. En d'autres termes pour $\forall \{t, t' | t \leq t'\}$ on a $d(p(i, j, t)) \leq d(p(i, j, t'))$.*

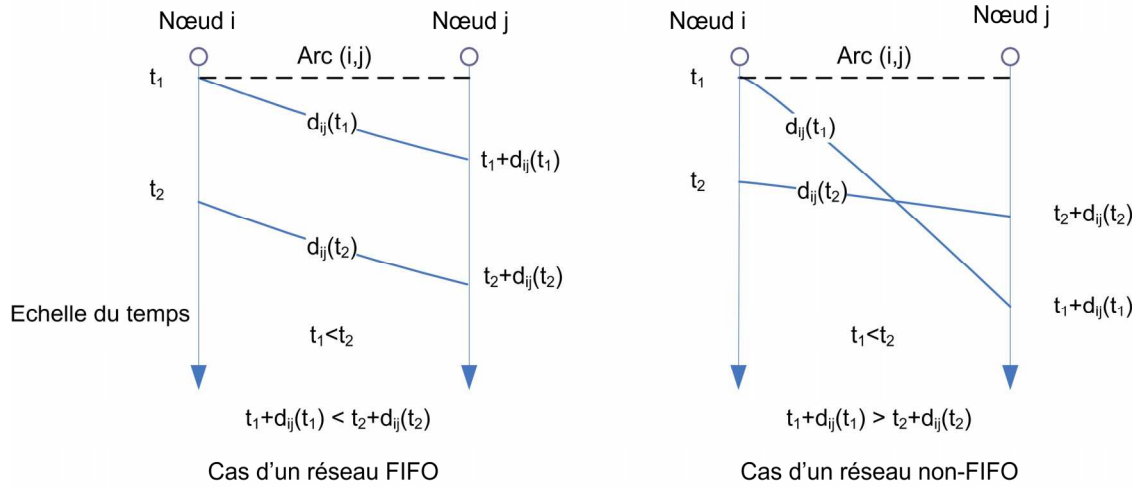


FIG. 3.8 – Propriété des réseaux FIFO

Lemme 4 Dans un réseau FIFO, si $p^*(i, j, t)$ est le plus court chemin (chemin le plus rapide) qui part de i à l'instant t vers un noeud j alors $p^*(i, j, t)$ est composé de plus courts chemins.

3.4.1.4 Temps de voyage minimum dans un réseau FIFO

Pour résoudre le problème du temps de voyage minimal, nous nous intéressons au problème du chemin le plus rapide $p_G^*(a, b, t_a)$ sur un graphe $G(N, E)$, qui part d'un noeud de départ a à un instant t_a , pour arriver au plus tôt à un noeud d'arrivée b . Si on note $\alpha_{aj}(t_a)$ le temps d'arrivée au plus tôt au noeud j en partant du noeud de départ a à un temps de départ t_a , le problème du chemin le plus rapide revient à retrouver les temps d'arrivée au plus tôt $\alpha_{aj}(t_a)$ pour tous les noeuds j de G . Les $\alpha_{aj}(t_a)$ sont solutions de l'équation suivante :

$$\alpha_{aj}(t_a) = \begin{cases} t_a & \text{si } j = a \\ \min_{i \in N^-(j)} \min_{t \geq \alpha_{ai}(t_a)} (t + d_{ij}(t)) & \text{sinon} \end{cases}$$

ce qui se traduit grâce à la propriété FIFO en :

$$\alpha(j, t_a) = \begin{cases} t_a & \text{si } j = a \\ \min_{i \in N^-(j)} (\alpha_{ai}(t_a) + d_{ij}(\alpha_{ai}(t_a))) & \text{sinon} \end{cases}$$

En effet dans le cas FIFO, on a $\min_{t \geq \alpha_{ai}(t_a)} (t + d_{ij}(t)) = \alpha_{ai}(t_a) + d_{ij}(\alpha_{ai}(t_a))$.

Ceci revient à résoudre le problème de plus court chemin classique d'un noeud de départ a vers tous les autres noeuds dans un réseau FIFO (PCC-FIFO).

L'algorithme de résolution est décrit par les étapes suivantes :

1. Initialisation :

$$\alpha_{ai}(t_a) = \infty \text{ et } prec(i, t_a) = \infty \text{ pour } \forall i \in N, i \neq a$$

$$\alpha_{aa}(t_a) = t_a, prec(a, t_a) = a \text{ et } N_c = \{a\}$$

2. Sélection d'un noeud :

$$\text{soit } i \text{ le noeud ayant } \min_{j \in N_c} (\alpha_{aj}(t_a))$$

$$N_c = N_c \setminus \{i\}$$

3. Exploration des successeurs possibles

Pour $\forall j \in N^+(i)$ Faire :

$$\left| \begin{array}{l} \text{si } \alpha_{aj}(t_a) > \alpha_{ai}(t_a) + d_{ij}(\alpha_{ai}(t_a)) \text{ alors Faire} \\ \alpha_{aj}(t_a) = \alpha_{ai}(t_a) + d_{ij}(\alpha_{ai}(t_a)) \\ prec(j, t_a) = i \\ \text{si } j \notin N_c \text{ alors } N_c = N_c \cup \{j\} \end{array} \right.$$

4. Critère de fin :

si $N_c \neq \emptyset$ alors aller à l'étape 2

sinon fin

FIG. 3.9 – Algorithme de plus court chemin dans les réseaux dynamiques FIFO

En effet, l'algorithme présenté est une adaptation du plus court chemin classique Dijkstra, qui permet de calculer à partir d'un noeud de départ donné, l'ensemble des plus courts chemins vers tous les autres noeuds, y compris le noeud d'arrivée souhaité. L'algorithme commence par initialiser les temps d'arrivée au plus tôt $\alpha_{ai}(t_a)$ des différents noeuds i , ainsi que leurs prédécesseurs $prec(i, t_a)$ à la valeur ∞ .

On définit ensuite l'ensemble des noeuds courants à explorer par l'algorithme. Cet ensemble est initialisé au départ au seul élément noeud de départ $N_c = \{a\}$. On initialise le temps d'arrivée minimal à ce noeud, ainsi que son précédent, avec $\alpha_{ai}(t_a) = t_a$ et $prec(a, t_a) = a$.

A partir de l'ensemble N_c , l'algorithme sélectionne un noeud i à explorer. Le noeud

sélectionné est celui qui offre la meilleure heure d'arrivée au plus tôt soit le min de $\alpha_{ai}(t_a)$ avec $i \in N_c$. Une fois le noeud i , sélectionné, il est retiré de l'ensemble des noeuds courants.

A partir du noeud sélectionné i , on explore tous les noeuds accessibles à partir de i ; il s'agit des noeuds éléments de $N^+(i)$. Pour chaque noeud j accessible à partir de i tel que $j \in N^+(i)$, on met à jour, si nécessaire, la valeur du temps d'arrivée au plus tôt $\alpha_{aj}(t_a)$ et du précédent $prec(j, t_a)$. En effet, si $\alpha_{aj}(t_a) > \alpha_{ai}(t_a) + d_{ij}(\alpha_{ai}(t_a))$ alors $\alpha_{aj}(t_a) = \alpha_{ai}(t_a) + d_{ij}(\alpha_{ai}(t_a))$ et $prec(j, t_a) = i$. Dans ce cas, le noeud j dont la valeur $\alpha_{aj}(t_a)$ a été améliorée sera inséré, s'il n'existe pas déjà, dans l'ensemble N_c des noeuds courants. Ce qui veut dire qu'il peut être sélectionné dans une étape ultérieure.

Ce procédé est répété tant qu'il existe encore des noeuds candidats ($N_c \neq \phi$). Ceci veut dire qu'on est déjà passé par une étape où on ne peut plus améliorer les temps d'arrivée au plus tôt des noeuds accessibles. Par ailleurs le fait de réinsérer dans l'ensemble N_c , des noeuds accessibles dont les valeurs d'arrivée au plus tôt ont été améliorées, permet d'explorer le graphe en tenant compte de la dimension temps : temps de départ possibles. Une fois le PCC-FIFO exécuté, le plus court chemin vers le noeud b est retrouvé par chaînage arrière en déterminant $prec(b, t_a), prec(prec(b, t_a)), t_a) \dots$ jusqu'à a .

3.4.1.5 Temps de voyage minimum dans un réseau non-FIFO

Pour résoudre le minimum de temps de voyage, nous nous intéressons, également dans les graphes non-FIFO, au problème du chemin le plus rapide $p^*(a, b, t_a)$, qui part d'un noeud de départ a à un instant t_a , pour arriver au plus tôt à un noeud d'arrivée b . Dans les graphes non-FIFO, la propriété FIFO n'est plus valable, d'où il devient nécessaire d'avoir plusieurs étiquettes, puisque dans ces graphes, on peut arriver au plus tôt au noeud b , par un chemin qui n'est pas forcément composé de plus courts chemins. Dans ce cas [Chabini et al.98], proposent de modifier l'algorithme précédent, en insérant dans l'ensemble N_c des couples noeud-temps (i, t) qui permettent de sauvegarder chaque arrivée possible au noeud i . De plus on introduit l'étiquette $prec(i, t)$ pour sauvegarder le précédent pour chaque arrivée possible au noeud i . Il s'agit d'une utilisation implicite du modèle espace-temps.

L'algorithme PCC-non-FIFO est décrit par les étapes suivantes :

1. Initialisation :

$\alpha_{ai}(t_a) = \infty$ et $prec(i, t) = (\infty, \infty)$ pour $\forall i \in N, i \neq a$ et $t \in TD(i, j)$ avec $j \in N^+(i)$

$\alpha_{aa}(t_a) = t_a, prec(a, t_a) = (a, t_a)$ et $N_c = \{(a, t_a)\}$

2. Sélection d'un noeud espace-temps :

soit (i, t) le noeud ayant $\min_{j \in N_c}(\alpha_{aj}(t))$

$N_c = N_c \setminus \{(i, t)\}$

3. Exploration des successeurs possibles

Pour $\forall j \in N^+(i)$ Faire :

	<i>si</i>	$(j, t + d_{ij}(t)) \notin N_c$	<i>alors</i>
		$N_c = N_c \cup \{(j, t + d_{ij}(t))\}$	
		$prec(j, t + d_{ij}(t)) = (i, t_a)$	
	<i>si</i>	$\alpha_{aj}(t_a) > t + d_{ij}(t)$	<i>alors</i> $\alpha_{aj}(t_a) = t + d_{ij}(t)$

4. Critère de fin :

si $(N_c \neq \emptyset$ et $t < \max(\alpha_{ai}(t_a) \forall i \in N)$ alors aller à l'étape 2

sinon fin

FIG. 3.10 – Algorithme de plus court chemin dans les réseaux dynamiques non FIFO

3.4.1.6 Modélisation du graphe dynamique et distribué

Dans le domaine du transport public, les réseaux des opérateurs peuvent être FIFO ou non FIFO. Dans les deux cas les SIAD qui leur sont associés permettent de calculer le chemin le plus rapide, celui qui permet d'arriver au plus tôt à une station j en partant d'une station i à un instant t , et ce en utilisant soit un algorithme PCC-FIFO pour les réseaux FIFO soit un algorithme non-FIFO pour les réseaux non-FIFO; i et j étant deux stations du réseau de l'opérateur de transport. D'autres SIAD peuvent aussi utiliser un modèle espace-temps, et résoudre ainsi le problème du plus court chemin dans le réseau dynamique, par un algorithme classique tel que celui de Dijkstra. Néanmoins, cette option procède à l'expansion du graphe initial, et augmente ainsi la complexité du problème. Enfin d'autres SIAD adoptent l'algorithme DOT (Decreasing Order of Time) détaillé par [Chabini97] qui permet pour

chaque départ possible à partir d'un noeud i de calculer les chemins les plus rapides vers tous les autres noeuds.

Sur un territoire donné, nous pouvons alors représenter la cartographie des SIAD par une juxtaposition d'agents d'information. Ceci peut être modélisé par un graphe distribué $G(N, E)$.

Nous associons à chaque AI une classe C . La $i^{ème}$ classe est notée C_i . C_i permet de résoudre le problème du chemin le plus rapide sur un graphe $G_i(N_i, E_i)$ où N_i et E_i sont respectivement l'ensemble de noeuds et l'ensemble des arcs relatifs à la classe C_i . Un noeud représente une station relative au réseau de transport associé à cette classe. Un arc représente une course réalisée par un moyen de transport relatif au même opérateur.

Ainsi un arc $(a, b) \in E_i$ si et seulement si $a \in N_i$ et $b \in N_i$ et le transport entre a et b est assuré par un moyen de transport géré par C_i . Le poids de l'arc (a, b) est noté $d_{ab}(t)$ puisque cette durée est fonction du temps de départ de la station a .

En effet nous associons, pour chaque arc (a, b) une liste de départs possibles planifiés par l'opérateur qu'on notera $TD(a, b) = (t_{d_1}(a, b), t_{d_2}(a, b), \dots, t_{d_k}(a, b), \dots, t_{d_{L(a,b)}}(a, b))$, avec $t_{d_k}(a, b)$ le k ème départ possible associé à l'arc (a, b) et $L(a, b)$ le nombre total de ces départs.

On notera aussi $(a, b)_i$, l'arc $(a, b) \in E_i$ ensemble des arcs de C_i . En effet si a et b font parti de plusieurs classes C_i , alors on peut avoir plusieurs possibilités d'arc (a, b) ou chaque arc représente un moyen de transport géré par une classe. Pour chaque arc $(a, b)_i$, on associe une liste d'horaires de départs possibles $TD(a, b)_i$ fournies par la classe C_i .

$\alpha_{Aa}(t_A)$ est défini comme l'heure d'arrivée au plus tôt à une station a à partir d'un temps de départ t_A de la station A .

En fin, on note G le graphe associé à l'ensemble de l'environnement distribué défini par n classes C_i . Il est noté :

$$G(N, E) \quad N = \bigcup_{i=1}^n N_i \quad E = \bigcup_{i=1}^n E_i$$

3.4.2 Principe du plus court chemin distribué et Horaire

Nous nous proposons dans la suite d'adapter l'approche présentée par [Wang et al.04] pour résoudre le problème de plus court chemin distribué, au problème du chemin le plus rapide de A vers B , c'est à dire celui qui permet d'arriver à un temps d'arrivée au plus tôt $\alpha_{AB}(t_A)$, et ce dans un environnement distribué et dynamique où la durée de l'arc (a, b) est une fonction $d_{ab}(t)$ du temps de départ t à la station a .

3.4.2.1 Graphe d'intersection

Pour ce faire, une première approche consiste à définir un graphe d'intersection étendu et horaire relatif à une requête donnée. Ce graphe noté $G_{\text{eint}_H(A,B,t_A)}$ est défini par :

- Un ensemble $N_{\text{eint}_H(A,B,t_A)}$ des noeuds d'intersections des différentes classes C_i ajouté aux noeuds associés respectivement à la station de départ A et la station d'arrivée B .
- Un ensemble d'arcs (a, b) de $E_{\text{eint}_H(A,B,t_A)}$, tel que $a \in N_{\text{eint}_H(A,B,t_A)}$ et $b \in N_{\text{eint}_H(A,B,t_A)}$. Chaque arc $(a, b)_i$ constitue un chemin local entre deux stations d'une même classe C_i . Pour déterminer le poids de l'arc $(a, b)_i$, nous devons considérer la liste $TD_{t_A}(a, b)_i$ qui constitue l'ensemble des départs de a vers b avec $t_{d_k}(a, b) > t_A$ pour tout $t_{d_k}(a, b) \in TD(a, b)_i$, et ce pour chaque classe C_i , contenant le couple $\{a, b\}$. Ensuite nous déterminons pour chaque arc (a, b) les chemins locaux les plus rapides $P_i^*(a, b, t_{d_k})$ correspondant aux différents départs de la liste $TD_{t_A}(a, b)_i$. En d'autres termes nous essayons de construire le modèle dépendant du temps du graphe d'intersection.

Ce graphe constitue le domaine de recherche du plus court chemin global $P_G^*(A, B, t_A)$.

Construire un tel graphe revient à créer une base de données intermédiaire et centralisée, qui, pour un noeud d'intersection a donné, va sauvegarder les chemins les plus rapides qui permettent d'aller de ce dernier vers tous les autres noeuds d'intersection qui lui sont accessibles, et ce pour chaque départ possible $t_{d_k}(a, b)$ de a . Les noeuds accessibles à partir de a sont tous les noeuds d'intersection b appartenant aux classes C_{i_a} qui contiennent a . Pour retrouver le plus court chemin global, on applique ensuite un algorithme de chemin le plus rapide sur ce graphe d'intersection dynamique.

Dans le procédé que nous présentons, relatif aux transports collectifs, la fréquence des départs possibles d'un noeud vers un autre, est très élevée. Par conséquent on ne peut pas

se permettre de construire un graphe d'intersection, qui va chercher toutes les informations relatives aux départs possibles, ainsi qu'aux durées des courses qui leurs correspondent et qui permettent d'aller d'un noeud d'intersection vers un autre.

Pour éviter la récupération et la mémorisation d'une telle quantité de données, nécessaire à l'exécution d'un algorithme de plus court chemin sur un graphe dynamique, nous proposons d'exécuter ce dernier à la volée, et de récupérer lors de son exécution toute donnée nécessaire à son fonctionnement.

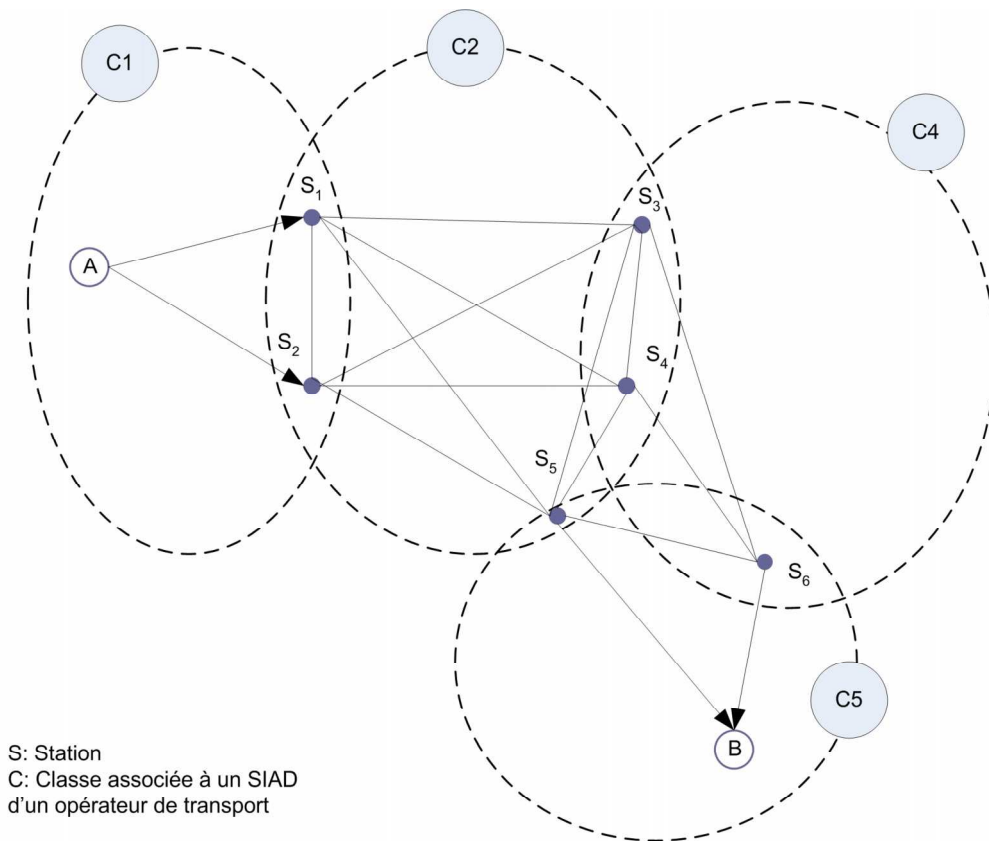


FIG. 3.11 – Exemple d'un graphe partiel virtuel

A partir de la requête $Req(A, B, t_A)$, nous définissons un graphe partiel virtuel $G_{vint(A,B)}$ comprenant un ensemble de noeuds comportant la station de départ A , la station d'arrivée B et les stations d'intersection du graphe G . Si a est un noeud de $G_{vint(A,B)}$, on note $I(a)$ l'ensemble des index i des classes contenant le noeud a tel que $I(a) = \{i | a \in C_i\}$. Les arcs de $G_{vint(A,B)}$ lient chaque noeud a à l'ensemble des noeuds b accessibles tels que $b \in \{C_i | i \in I(a)\}$. Ainsi les seules informations dont on dispose sont l'ensemble $N_{vint(A,B)}$ des noeuds de $G_{vint(A,B)}$, et les classes aux quelles appartient chaque noeud de cet ensemble. Les arcs de

$E_{vint(A,B)}$ représentent les liaisons possibles entre ces différents noeuds.

3.4.2.2 Algorithme

Une question s'impose : Comment exécuter un algorithme de plus court chemin sur un graphe d'intersection dynamique qui n'est pas encore construit ? Et quel algorithme utiliser si on ne connaît pas, à l'avance, la nature de ce graphe d'intersection (FIFO ou non FIFO).

Certes, le graphe dynamique n'est pas totalement construit, puisque les poids des arcs, ainsi que les différents départs possibles à partir de chaque noeud restent inconnus au départ. Mais c'est lors de l'exécution de l'algorithme du chemin le plus rapide, sur ce graphe, qu'on va récupérer, via l'envoi de requêtes locales du type $req(a, b, t_a)$, les poids des arcs. Ces requêtes permettent de mettre à jour l'heure d'arrivée au plus tôt à chaque sommet et par conséquent le précédent de chaque sommet. Le graphe $G_{vint(A,B)}$ ne présente en effet qu'un ensemble de possibilités d'exploration : un plan pour l'envoi des requêtes.

Les arcs de $G_{vint(A,B)}$ relient entre A , B et les stations d'intersection. Soient a et b deux noeuds de $G_{vint(A,B)}$. Pour retrouver $\alpha_{ab}(t_a)$ heure d'arrivée au plus tôt à b en partant de a à l'instant t_a , on lance la requête locale $req(a, b, t_a)$ à la classe C_i qui contient l'arc (a, b) . Cette requête va lancer l'exécution d'un algorithme du chemin le plus rapide par la classe C_i . Selon le réseau géré par la classe, ce dernier peut être soit un algorithme PCC-FIFO, soit un algorithme PCC-non-FIFO. Mais dans tous les cas, cette requête retourne l'heure d'arrivée du chemin le plus rapide : celui qui permet, à partir d'un temps d'arrivée t_a à la station a d'arriver au plus tôt à la station d'arrivée b . En effet t_a c'est l'horaire de départ demandé. L'horaire de départ effectif est défini par $\alpha_{ab}(t_a) - d(P_i^*(a, b, t_a))$. Et la durée globale du trajet est définie par $d_{ab}(t_a) = \alpha_{ab}(t_a) - t_a$.

Si a et b sont deux noeuds d'intersection, deux cas sont à distinguer :

- (a, b) est un arc d'une seule classe C_i . La requête $req(a, b, t_a)$ retourne le temps d'arrivée au plus tôt à b en partant de a à t_a , soit $\alpha_{ab}(t_a)$.

La durée $d_{ab}(t_a) = \alpha_{ab}(t_a) - t_a$. Pour vérifier si (a, b) est FIFO, il faut montrer que

pour $\forall\{t, t' | t \leq t'\}$ on a $t + d_{ab}(t) \leq t' + d_{ab}(t')$. Ce qui revient à montrer que pour

$$\forall\{t, t' | t \leq t'\} \quad \text{on a} \quad \alpha_{ab}(t) \leq \alpha_{ab}(t')$$

Pour le montrer, on suppose qu'il existe t et t' tels que $t \leq t'$ et $\alpha_{ab}(t) > \alpha_{ab}(t')$. Ceci veut dire que l'heure d'arrivée de a vers b en partant de a à t est $\alpha_{ab}(t')$. Ce qui est absurde, puisque $req(a, b, t_a)$ est sensé retourner l'heure d'arrivée au plus tôt $\alpha_{ab}(t)$. D'où on montre que $\forall\{t, t' | t \leq t'\}$ on a $\alpha_{ab}(t) \leq \alpha_{ab}(t')$, ce qui veut dire que l'arc (a, b) du graphe $G_{vint(A,B)}$ vérifie la propriété FIFO.

- S'il existe différents arcs $(a, b)_i$ correspondant à différentes classes C_i , suite à ce qui a été démontré précédemment, chaque arc $(a, b)_i$ vérifie la propriété FIFO. Cela n'empêche pas d'avoir deux arcs $(a, b)_1$ et $(a, b)_2$ tels qu'il existe $t < t'$ et $\alpha_{ab_1}(t) > \alpha_{ab_2}(t')$. Pour que (a, b) puisse vérifier aussi la propriété FIFO, le temps d'arrivée au plus tôt en b , en partant de a à l'instant t_a sera défini comme le minimum des temps d'arrivées au plus tôt des différents arcs $(a, b)_i$ relatifs aux différentes classes C_i contenant (a, b) . Ainsi :

$$\alpha_{ab}(t_a) = \min_{i \in I(a,b)} (\alpha_{ab_i}(t_a))$$

La durée du trajet étant $d_{ab}(t_a) = \alpha_{ab}(t_a) - t_a$, on peut montrer comme précédemment que (a, b) vérifie la propriété FIFO. Néanmoins il est nécessaire dans ce cas de sauvegarder par quel réseau on est passé, pour pouvoir retrouver le chemin. Nous noterons $r(b)$ la classe retenue, pour accéder à b , à partir de $prec(b, t_A)$. Ce dernier étant le précédent de b , qui permet d'arriver au plus tôt à b .

Nous démontrons alors que tout arc (a, b) de $G_{vint(A,B)}$ vérifie la propriété FIFO. De ce fait, par la définition même des graphes dynamiques FIFO, le graphe $G_{vint(A,B)}$ est FIFO.

L'algorithme qui sera exécuté à la volée est une adaptation distribué du PCC-FIFO. Ce PCCD-FIFO est décrit par la figure 3.12.

L'algorithme de plus court chemin horaire et distribué (PCCDH) commence par construire le graphe d'intersection virtuel $G_{vint(A,B)}$. Il exécute la dessus le PCCD-FIFO. Le chemin obtenu est le $P_{G_{vint(A,B)}}^*(A, B, t_A) = \{s_1 - \dots - s_k - s_{k+1} - \dots - s_m\}$ avec $s_1 = A$, $s_m = B$, les s_k étant les noeuds du graphe $G_{vint(A,B)}$. On déterminera ensuite le détail des chemins respec-

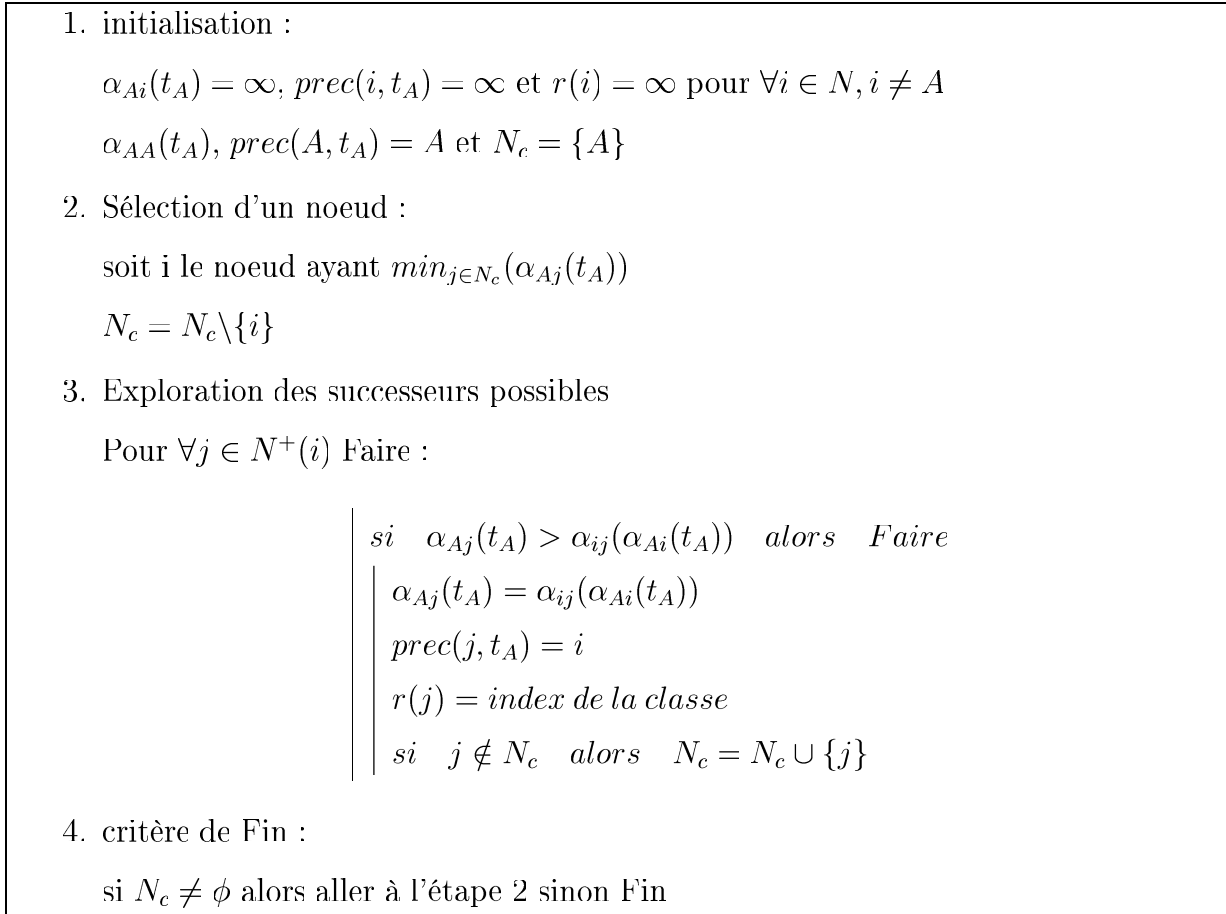


FIG. 3.12 – Algorithme de plus court chemin distribué-FIFO

tifs $p_k^*(s_k, s_{k+1}, \alpha_{As_k}(t_A))$ aux différentes classes C_i retenues. Ces chemins constitueront l'offre globale $P_G^*(A, B, t_A)$.

Lemme 5 *Un chemin $P_{G_{vint(A,B)}}^*(A, B, t_A)$ de $G_{vint(A,B)}$ qui permet d'arriver au plus tôt à B à l'instant $\alpha_{AB}(t_A)$ correspond à un chemin $P_G(A, B, t_A)$ sur $G(N, E)$ avec la même heure d'arrivée.*

Lemme 6 *S'il existe un chemin $P_G^*(A, B, t_A)$, le plus rapide sur $G(N, E)$ alors il existe un chemin $P_{G_{vint(A,B)}}^*(A, B, t_A)$, qui contient les noeuds d'intersection contenus dans $P_G^*(A, B)$, et qui offre la même heure d'arrivée au plus tôt à B qu'on note $\alpha_{AB}(t_A)$.*

Preuve Le Lemme 6 peut être démontré ainsi : En effet Si $P_G^*(A, B, t_A)$ est le chemin le plus rapide sur $G(N, E)$, alors il passe obligatoirement de A vers B par des noeuds d'intersection ($P_G^*(A, B, t_A) = \{a = s_1 - \dots - s_n = B\}$). Par ailleurs, $P_G^*(A, B, t_A)$ est composé d'offres

1. Construire le $G_{vint(A,B)}$ associé au $G(N, E)$.
2. Déterminer $P_{G_{vint(A,B)}}^* = \{s_1 - \dots - s_m\}$ en exécutant PCCD-FIFO.
3. Pour $k = 1$ à $k = m - 1$
 - (a) pour (s_k, s_{k+1}) déterminer la classe C_i relative à $r(s_{k+1})$
 - (b) pour (s_k, s_{k+1}) déterminer le détail de $p_i^*(s_k, s_{k+1}, \alpha_{As_k}(t_A))$ en envoyant la requête $req(s_k, s_{k+1}, \alpha_{As_k}(t_A))$ à la classe C_i
4. les $p_i^*(s_k, s_{k+1})$ constituent le chemin global optimal $P_G^*(A, B, t_A)$

FIG. 3.13 – Algorithme de plus court chemin distribué et Horaire

locales, $p(A, s_2, t_A)$, $p(s_2, s_3, \alpha_{As_2}(t_A)) \dots p(s_n, b, \alpha_{As_n}(t_A))$ qui partent tous à partir d'un temps $t \geq t_A$. Or, ces éléments font partie du domaine de recherche de $G_{vint(A,B)}$, par définition.

Donc il existe un chemin $P_{G_{vint(A,B)}}^*(A, B)$ qui correspond à $P_G^*(A, B)$ qui passe par les mêmes noeuds d'intersection et offre la même heure d'arrivée au plus tôt.

En se basant sur ces deux lemmes, nous démontrons le théorème suivant :

théorème 2 *l'algorithme PCCDH calcule le chemin le plus rapide $P_G^*(A, B, t_A)$ sur le graphe $G(N, E)$ qui permet à partir d'une heure d'arrivée t_A à la station A d'arriver au plus tôt à B à l'instant $\alpha_{AB}(t_A)$.*

Preuve soit $P_{G_{vint(A,B)}}^*(A, B, t_A)$ le chemin le plus rapide de A à B sur le graphe $G_{vint(A,B)}$ calculé à l'étape 2 du PCCDH. D'après le Lemme 5, ce chemin correspond à un chemin $P_G^*(A, B, t_A)$ sur le graphe $G(N, E)$ qui offre une même heure d'arrivée au plus tôt $\alpha(A, B, t_A)$.

S'il existe un autre chemin sur $G(N, E)$ qui est plus rapide que $P_G^*(A, B, t_A)$ et qui offre une heure d'arrivée $\alpha'_{AB}(t_A)$ telle que $\alpha'_{AB}(t_A) < \alpha_{AB}(t_A)$, alors d'après le Lemme 6, il doit exister un chemin qui lui correspond sur $G_{vint(A,B)}$ présentant la même heure d'arrivée. Ceci est en contradiction avec le fait que $P_{G_{vint(A,B)}}^*(A, B)$ est plus court chemin de A à B sur le graphe $G_{vint(A,B)}$.

Par conséquent $P_G^*(A, B)$ obtenu à l'étape 3, par rapport à $P_{G_{vint(A,B)}}^*(A, B, t_A)$, est le chemin le plus rapide de A vers B sur le graphe $G(N, E)$.

Dans le chapitre suivant nous présenterons un exemple d'exécution du PCCDH.

3.4.2.3 PCCDH et organisation SMA

On associe chaque SIAD d'un opérateur à un agent d'information (AI), où chaque AI correspond à une classe. L' AI_i est capable de répondre à une requête locale d'itinéraire $req(a, b, t_a)$ par le chemin le plus rapide $p_i^*(a, b, t_a)$ qui permet de partir en a à t_a et d'arriver le plus tôt possible en b , et ce sur un réseau de transport modélisé par un graphe G_i .

Pour répondre à une requête globale $Req(A, B, t_A)$ sur le graphe global horaire et distribué $G(N, E)$, l'algorithme PCCDH sera exécuté par un agent médiateur.

L'agent médiateur ou courtier (AC) commence alors par construire le graphe d'intersection virtuel $G_{vint(A,B)}$ correspondant à la requête $Req(A, B, t_A)$, en demandant pour chaque noeud les noeuds qui lui sont accessibles. Le médiateur exécute ensuite l'algorithme PCCD-FIFO sur ce dernier pour déterminer la meilleure offre globale qui répond à $Req(A, B, t_A)$. Une fois le chemin le plus rapide déterminé (ensemble d'offres locales), cet agent va demander le détail de chaque offre locale retenue. La composition et la recherche de l'itinéraire sont assurées par cet algorithme : compétence de l'agent AC.

Dans cette composition, on suppose que l'agent AC a une vision globale de tous les AI de l'environnement. Le domaine de recherche n'est pas limité ; il est associé à l'ensemble des AI qui définissent l'environnement. Dans ce cas, l'organisation Interface-Médiateur-Adaptateur est celle qui s'adapte au mieux pour utiliser un tel algorithme, puisque le médiateur n'a pas besoin de limiter son domaine de recherche.

3.5 Algorithme de sélection du domaine de recherche

Les algorithmes de recherche et de composition d'itinéraires présentés à ce stade -PCCD ou PCCDH- permettent de composer un plus court chemin global sur un environnement distribué à partir de différentes offres locales sur un ensemble de SIAD qui partagent des stations d'intersection.

Nous proposons de modéliser cet environnement par un graphe d'adjacence non-orienté $G_e(N_e, E_e)$. Les noeuds éléments de N_e représentent l'ensemble des agents AI_i associés aux différents opérateurs de transports. Les arcs éléments E_e représentent les possibilités de

passages d'un réseau d'un opérateur vers un autre. En d'autres termes un arc représente l'existence d'un ou plusieurs pôles d'échanges entre deux AI. Les arcs sont tous valués d'un poids unitaire. Dans le SICM, le $G_e(N_e, E_e)$ est constitué par tous les AI inscrits auprès de l'agent annuaire sélectionneur AS.

Nous définissons deux AI comme adjacents s'ils correspondent à des noeuds adjacents sur ce graphe.

Si, pour une requête $Req(A, B, t_A)$ ou $Req(A, B)$, AI_A correspond à l'agent d'information associé à la station A , et que AI_B est celui associé à la station B , alors les différents chemins de G_e qui permettent de relier AI_A et AI_B constituent l'ensemble des séquences de réseaux de transport qu'un utilisateur pourrait emprunter pour aller de A vers B .

Par ailleurs la complexité des algorithmes de composition d'itinéraire proposés précédemment, dépend essentiellement du nombre des noeuds des graphes d'intersection utilisés. En effet pour le PCCD cette complexité est de $O(|N_{cint(A,B)}|^2)$. Pour le PCCDH, elle est de $O(|N_{vint(A,B)}|^2)$.

Pour minimiser le temps de réponse à une requête globale du type $Req(A, B)$, il est nécessaire de minimiser le nombre d'intersections. Ceci revient à minimiser le nombre d'AI à impliquer dans la composition de l'itinéraire global.

De ce fait, le domaine de recherche $D_{R(A,B)}$ associé à la requête $Req(A, B)$ est souvent défini par :

- l'AI correspondant à la station de départ,
- celui associé à la station d'arrivée,
- et un ensemble d'AI intermédiaires qui assurent le lien entre les deux premiers.

Nous proposons de représenter le $D_{R(A,B)}$ comme un sous graphe $D_{R(A,B)}(N_{D_{R(A,B)}}, E_{D_{R(A,B)}}) \subset G_e(N_e, E_e)$. Il s'agit d'un sous-graphe d'adjacence d'agents d'information qui contient au moins un chemin d'agents adjacents qui lie AI_A et AI_B , et tel que $N_{D_{R(A,B)}} \subset N_e$ et $E_{D_{R(A,B)}} \subset E_e$.

Pour construire et limiter ce domaine de recherche $D_{R(A,B)}$, nous nous proposons d'utiliser l'un des deux algorithmes suivants, et qui permettent de limiter le nombre de chemins à considérer sur $G_e(N_e, E_e)$ pour lier AI_A et AI_B .

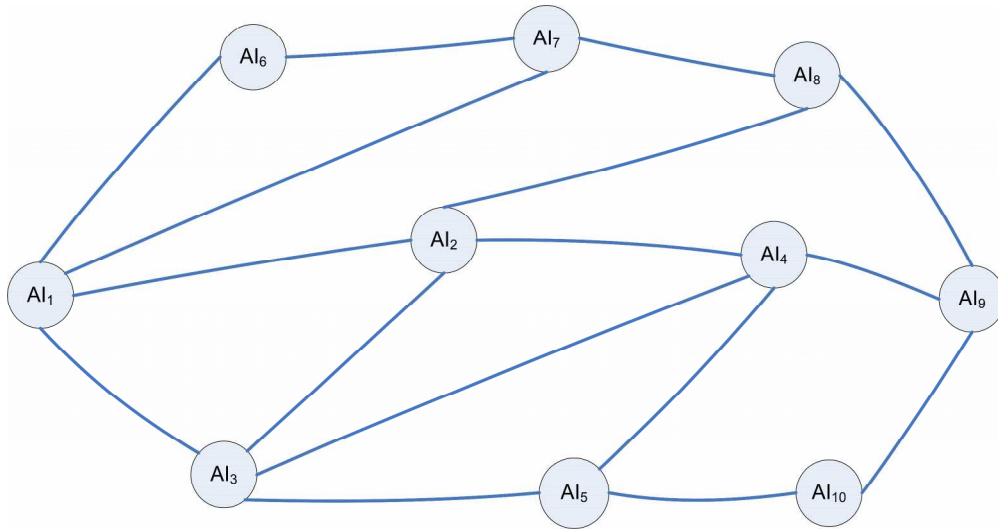


FIG. 3.14 – Exemple d’un graphe d’adjacence d’AI

Nous proposons d’utiliser l’algorithme de k plus courts chemins [Eppstein94], [Martins et al.98]. Cet algorithme permet de rechercher sur le graphe d’adjacence global $G_e(N_e, E_e)$, les k plus courts chemins qui correspondent aux k plus petits domaines de recherche, en termes du nombre d’agents d’informations à impliquer, et qui permettent de construire une offre globale. L’algorithme de plus courts chemins, est basé sur une heuristique de pénalité. Il applique l’algorithme Dijkstra pour calculer le plus court chemin, ensuite il modifie le graphe en imposant une augmentation des poids des arcs du plus court chemin. Il applique ensuite Dijkstra pour trouver le 2^{eme} plus court chemin, et ainsi de suite jusqu’au k^{eme} plus court chemin.

Dans la littérature, nous trouvons par ailleurs un algorithme récursif dit d’inondation et qui permet de trouver tous chemins qui relient entre deux noeuds. Ce dernier est très utilisé dans les problématiques de routage dans les réseaux informatiques, mais aussi dans la découverte de routes entre deux noeuds dans les réseaux dit Ad Hoc [Costa et al.03] : réseaux dont la topologie est dynamique et change en fonction des terminaux mobiles qui s’y connectent. Le principe de cet algorithme est de partir d’un noeud d’origine AI_A , d’explorer tous ces successeurs et de sauvegarder en parallèle les chemins explorés, et ce en évitant d’explorer les noeuds déjà retenus dans chacun de ces chemins. L’algorithme récursif s’appelle lui même pour chaque étape de l’exploration d’un noeud courant AI_i . L’exploration est

arrêtée quand le successeur de AI_i est le noeud destination AI_B : le chemin obtenu dans ce cas est une solution à retenir.

Nous modifions cet algorithme pour qu'il arrête l'exploration quand il aura trouvé tous les chemins constitués au plus, de k arcs, ce qui correspond sur le graphe d'adjacence aux nombre maximal souhaité de transbordements inter-opérateurs. L'algorithme est illustré par les étapes suivantes :

1. initialisation :

- AI_A noeud de départ
- AI_B noeud d'arrivée
- p est un chemin vide

2. *inondation* (AI_A, AI_B, p)

- ajouter AI_A au chemin p
- si ($(AI_A = AI_B)$ et (nombre d'arcs de $p \leq k$)) alors p est une solution.
- pour $\forall AI_i \in N^+(AI_A)$ Faire :

si	AI_i n'est pas un noeud de p
	$inondation(AI_i, AI_B, p)$

$N^+(AI_i)$: ensemble des noeuds successeurs de AI_i

FIG. 3.15 – Algorithme d'inondation modifié

l'algorithme de k plus courts chemins et celui de l'inondation modifié (au maximum k transbordements) sont deux algorithmes candidats à la sélection du domaine de Recherche, compétence de l'agent annuaire sélectionneur (AS). Les deux algorithmes retournent un ensemble restreint de n chemins qui vont nous permettre de délimiter et de sélectionner le domaine de recherche $D_R(A, B)$.

Une fois ces chemins déterminés sur G_e , 3 approches sont à considérer pour définir le domaine de recherche :

- Un domaine de recherche $D_R(A, B)_n$ de rang n , est associé à chaque chemin $\{AI_A - \dots - AI_i - \dots - AI_B\}$ de rang n . L'AC applique une composition d'itinéraire global sur chaque $D_R(A, B)_n$. Il fournit à l'utilisateur n offres d'itinéraires globaux.

- Un domaine de recherche $D_R(A, B)_n$ est associé à chaque plus court chemin $\{AI_A - \dots AI_i \dots - AI_B\}$ de rang n . L'AC présentent à l'utilisateur les n possibilités de composition. Ensuite il applique une composition d'itinéraire global sur le $D_R(A, B)_n$ correspondant à l'ensemble des opérateurs choisis par l'opérateur.
- Un domaine de recherche $D_R(A, B)$ est formé par tous les AI inclus dans l'ensemble des n chemins sélectionnés. L'AC assure la composition de l'itinéraire global sur le $D_R(A, B)$.

Certes, l'itinéraire global et optimal recherché sur $D_R(A, B)$, peut être différent de celui recherché sur $G_e(N_e, E_e)$. Néanmoins dans le contexte du transport public, l'aménagement des lignes de transport et la cartographie des zones géographiques associés aux réseaux de transport qui sont eux mêmes associés à des communes ou à des agglomérations, font que les chemins les plus rapides entre deux stations (ou les moins chers) sont souvent ceux qui impliquent un minimum d'opérateurs de transports et par conséquent un minimum de pôles d'échanges. En effet entre deux SIAD régionaux il y a souvent un SIAD national. Entre deux SIAD communaux il y a souvent un SIAD régional.

3.6 Conclusion

Dans ce chapitre nous avons présenté un algorithme de plus court chemin distribué PCCD et proposé un algorithme de plus court chemin distribué et horaire PCCDH. Ces algorithmes ont été retenus comme algorithmes de recherche et de composition d'itinéraires (ACRI). l'ACRI est exécuté par le médiateur dans le SICM pour explorer à la volée les offres locales des différents AI, et composer ainsi en ligne une offre d'itinéraire global qui correspond au critère de recherche de l'utilisateur.

Par ailleurs, comme algorithme de sélection du domaine de recherche (ASDR,) nous avons proposé d'utiliser soit un algorithme de k plus court chemin, soit un algorithme d'inondation modifié, et ce sur un graphe d'adjacence qui définit la cartographie des réseaux de transport. l'ASDR est exécuté par l'AS, agent annuaire sélectionneur, et ce pour réduire le domaine de recherche et le faire correspondre, au mieux, à la requête de l'utilisateur.

Pour illustrer les différents algorithmes présentés, nous présentons dans le chapitre sui-

vant, différents scénarios de simulation qui permettront de détailler : l'exécution des algorithmes précédents, et le contenu des messages échangés entre les agents du SICM pour réaliser cette composition d'itinéraire.

Nous présenterons aussi des solutions techniques qui permettront d'implémenter le SICM dans un contexte industriel.

Chapitre 4

Implémentation et scénarios de simulation

4.1 Introduction

Le Système d'Information Coopératif de Mobilité (SICM) est composé d'agents d'information qui coopèrent entre eux pour rechercher et composer en ligne, d'une manière dynamique, une information d'aide au déplacement multimodal dispersée sur un ensemble de SIAD associés à différents opérateurs de transport.

Après avoir exposé les algorithmes retenus pour la sélection du domaine de recherche (ASDR) et pour la composition de l'itinéraire multimodal et multi-opérateurs (ACRI), nous nous proposons dans ce dernier chapitre, d'exposer les scénarios de simulation de ces algorithmes, ce qui nous permettra par la suite de détailler le fonctionnement global du SICM. Nous détaillons, en effet, le contenu des messages entre les différents agents du SICM. En effet, ce sont les communications et les coopérations inter-agents qui vont permettre au SICM de répondre à une requête d'un itinéraire global, en exploitant les compétences individuelles des différents agents mis en oeuvre. L'AS assure la sélection du domaine de recherche et définit les AI à interroger. L'AC assure la composition dynamique de l'itinéraire global en faisant appel aux AI concernés. Les AI communiquent avec les SIAD des différents opérateurs de transport pour répondre à des requêtes d'itinéraires dites locales.

Une fois les algorithmes simulés et les données à échanger entre agents définies, nous étudions les scénarios de déploiement et d'implémentation du SICM, en fonction du contexte

de l'application. Il s'agit là, de présenter concrètement un prototype d'information coopératif de mobilité, qui met en oeuvre les technologies d'intégration de l'information et celles de la programmation orientée agents, et qui peut être diffusable industriellement.

4.2 Implémentation et Simulation des algorithmes du SICM

4.2.1 Scénarios de simulation

Notre première motivation dans ce travail est de montrer la faisabilité d'un calcul d'itinéraire global et distribué pour les opérateurs de transport Français. Or, il est encore plus difficile de montrer cette faisabilité quand on n'a pas l'accès aux différents systèmes d'information et d'aide aux déplacements associés aux opérateurs de transports. Ainsi pour nos travaux de recherche, nous avons basé la simulation sur des exemples et des données théoriques, et qui restent conformes au contexte réel des transports publics.

Par ailleurs, nous avons fait le choix, d'utiliser le langage de programmation JAVA. En effet, ce langage orienté objet assure la portabilité du code sur toute plateforme informatique, puisque ce dernier s'exécute toujours sur une machine virtuelle JAVA JVM (Java Virtual Machine) qui sera adaptée au matériel informatique (ordinateur, téléphone portable, PDA). De plus, il existe aujourd'hui une panoplie de bibliothèques Java libres d'accès qui offrent diverses fonctionnalités, et notamment la programmation des agents (plateformes multi-agents) et l'implémentation des services web : technologies que nous avons utilisés pour l'implémentation du SICM.

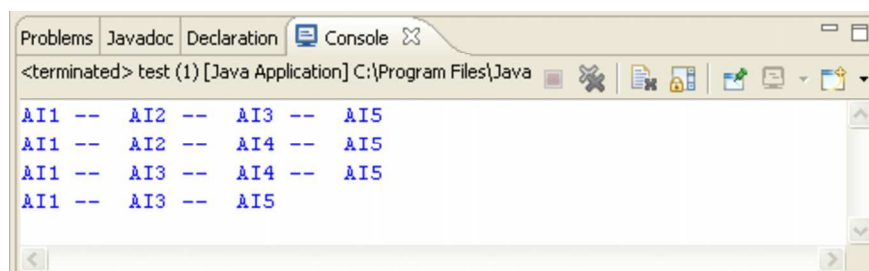
Nous exposons dans cette partie deux exemples d'exécution des algorithmes utilisés : l'ASDR et l'ACRI.

4.2.1.1 Exemple ASDR

Pour illustrer l'algorithme de sélection du domaine de recherche (ASDR), nous considérons le graphe d'adjacence de la figure 3.15 du chapitre 3. Il s'agit d'un graphe qui représente l'ensemble des agents inscrits auprès de l'agent annuaire sélectionneur (AS). Dans le SICM ce dernier détermine et limite, en fonction de la requête de l'utilisateur $Req(A, B)$, le domaine de recherche de l'agent courtier compositeur (AC). Il fournit à ce dernier un sous graphe d'adjacence avec l'ensemble des stations d'intersection qui lui sont associées.

Pour ce faire, l'AS commence par déterminer les agents AI_A et AI_B associés respectivement aux stations : A de départ et B d'arrivée. Ensuite en exécutant l'ASDR sur le graphe d'adjacence, l'agent annuaire détermine un ensemble restreint de n chemins qui permettent sur ce graphe d'aller de AI_A vers AI_B .

Pour cet exemple nous supposons que la station A de départ est associée à l'agent AI_1 et que celle d'arrivée B est associée à AI_5 . L'exécution de l'algorithme d'inondation modifié, paramétré à un nombre de transbordement maximal $k_T = 3$, permet à l'AS de sélectionner n chemins, avec $n = 4$ (voir Fig 4.1), sur un ensemble de 37 chemin qui mènent de AI_A vers AI_B (voir Fig 4.2).



```
<terminated> test (1) [Java Application] C:\Program Files\Java
AI1 -- AI2 -- AI3 -- AI5
AI1 -- AI2 -- AI4 -- AI5
AI1 -- AI3 -- AI4 -- AI5
AI1 -- AI3 -- AI5
```

FIG. 4.1 – Exécution de l'algorithme d'inondation avec $k_T = 3$

Les chemins sélectionnés impliquent au plus 5 agents d'information : AI_1 , AI_2 , AI_3 , AI_4 , et AI_5 (voir Fig 4.6). Ces agents représentent les SIAD illustrés par la figure 4.3.

Pour définir le domaine de recherche $D_R(A, B)$ associé à la requête $req(A, B)$, l'AS peut adopter deux approches différentes :

- La première approche consiste à construire 7 domaines de recherche correspondants aux 7 chemins sélectionnés. Dans ce cas l'AC compose 7 itinéraires globaux en appliquant l'algorithme de plus court chemin distribué (PCCD) sur chaque domaine de recherche. Ainsi l'itinéraire global associé au domaine $AI_1 - AI_3 - AI_5$ présente une durée de 170 et celui associé au domaine $AI_1 - AI_2 - AI_4 - AI_5$ dure 180 (Voir Fig 4.4 et Fig 4.5).
- La deuxième approche consiste à sélectionner tous les agents impliqués dans les différents chemins, et tous les arcs qui, dans le graphe d'adjacence les lient deux à deux. L'AS sélectionne ainsi un sous-graphe qui détermine le seul domaine de recherche associé à $req(A, B)$. l'exécution du PCCD sur ce domaine de recherche $D_R(A, B)$ permet d'obtenir un plus court chemin global de durée 170. Il s'agit en effet du même chemin retrouvé sur le domaine $AI_1 - AI_3 - AI_5$.

```

<terminated> test (1) [Java Application] C:\Program Files\Java\jre1.5.0_06\bin\javaw.exe
AI1 -- AI2 -- AI3 -- AI4 -- AI5
AI1 -- AI2 -- AI3 -- AI4 -- AI9 -- AI10 -- AI5
AI1 -- AI2 -- AI3 -- AI5
AI1 -- AI2 -- AI4 -- AI3 -- AI5
AI1 -- AI2 -- AI4 -- AI5
AI1 -- AI2 -- AI4 -- AI9 -- AI10 -- AI5
AI1 -- AI2 -- AI8 -- AI9 -- AI4 -- AI3 -- AI5
AI1 -- AI2 -- AI8 -- AI9 -- AI4 -- AI5
AI1 -- AI2 -- AI8 -- AI9 -- AI10 -- AI5
AI1 -- AI3 -- AI2 -- AI4 -- AI5
AI1 -- AI3 -- AI2 -- AI4 -- AI9 -- AI10 -- AI5
AI1 -- AI3 -- AI2 -- AI8 -- AI9 -- AI4 -- AI5
AI1 -- AI3 -- AI2 -- AI8 -- AI9 -- AI10 -- AI5
AI1 -- AI3 -- AI4 -- AI2 -- AI8 -- AI9 -- AI10 -- AI5
AI1 -- AI3 -- AI4 -- AI5
AI1 -- AI3 -- AI4 -- AI9 -- AI10 -- AI5
AI1 -- AI3 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI9 -- AI4 -- AI2 -- AI3 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI9 -- AI4 -- AI3 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI9 -- AI4 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI9 -- AI10 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI2 -- AI3 -- AI4 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI2 -- AI3 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI2 -- AI4 -- AI3 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI2 -- AI4 -- AI5
AI1 -- AI6 -- AI7 -- AI8 -- AI2 -- AI4 -- AI9 -- AI10 -- AI5
AI1 -- AI7 -- AI8 -- AI9 -- AI4 -- AI2 -- AI3 -- AI5
AI1 -- AI7 -- AI8 -- AI9 -- AI4 -- AI3 -- AI5
AI1 -- AI7 -- AI8 -- AI9 -- AI4 -- AI5
AI1 -- AI7 -- AI8 -- AI9 -- AI10 -- AI5
AI1 -- AI7 -- AI8 -- AI2 -- AI3 -- AI4 -- AI5
AI1 -- AI7 -- AI8 -- AI2 -- AI3 -- AI5
AI1 -- AI7 -- AI8 -- AI2 -- AI3 -- AI5
AI1 -- AI7 -- AI8 -- AI2 -- AI4 -- AI3 -- AI5
AI1 -- AI7 -- AI8 -- AI2 -- AI4 -- AI5
AI1 -- AI7 -- AI8 -- AI2 -- AI4 -- AI9 -- AI10 -- AI5

```

FIG. 4.2 – Exécution de l’algorithme d’inondation

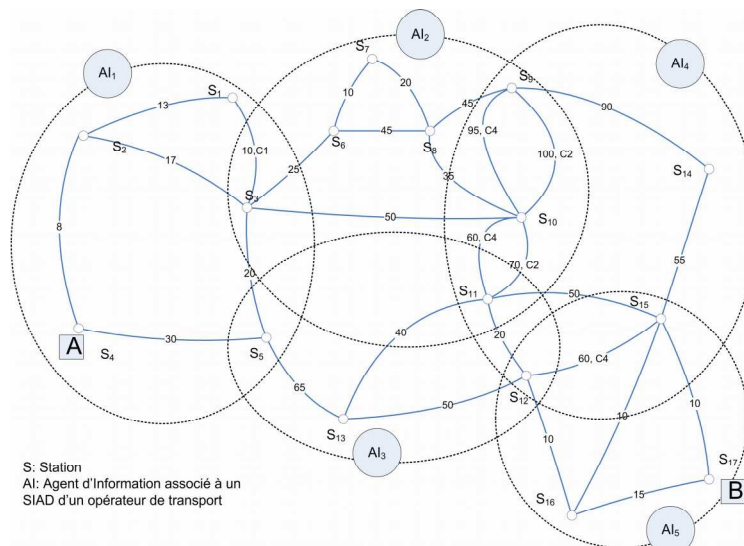


FIG. 4.3 – Ensemble des réseaux des SIAD associés au domaine de recherche

La première approche permet d’offrir différents itinéraires composés à l’utilisateur. Mieux encore, l’utilisateur peut choisir, avant même de commencer la composition, la succession des

opérateurs qui lui convient. Tous les chemins obtenus, auront un nombre de transbordements inférieur ou égal à la limite souhaitée. La deuxième approche permet seulement de limiter le domaine de recherche. L'utilisateur n'est plus assuré d'avoir au plus k_T transbordements par chemin, puisque d'autres adjacences entre les AI retenus sont prises en compte dans le graphe $D_R(A, B)$: ce qui permet de créer des chemins ayant un nombre de transbordements supérieur à k_T .

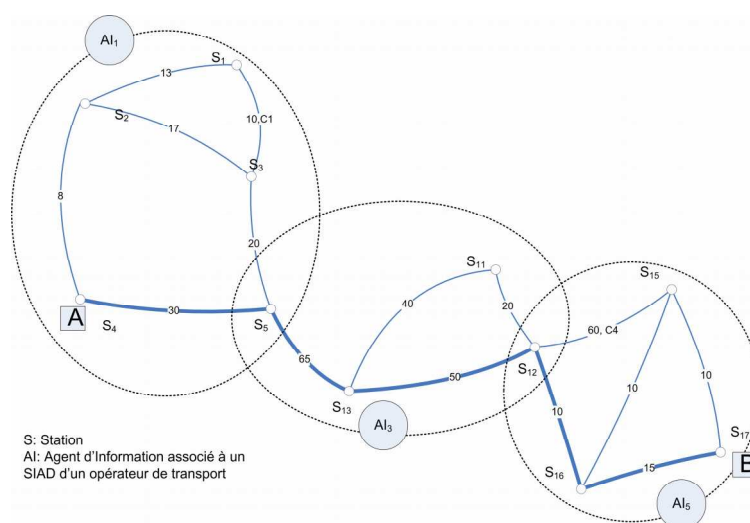


FIG. 4.4 – Itinéraire global relatif au domaine de recherche $AI_1 - AI_3 - AI_5$

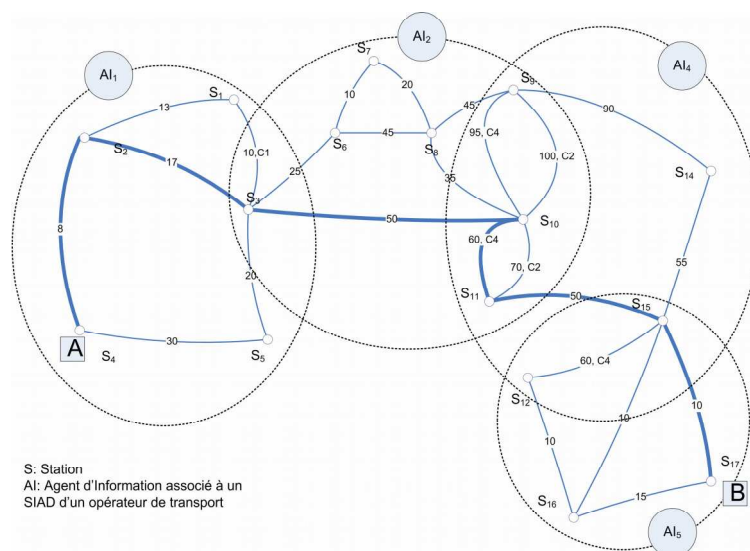


FIG. 4.5 – Itinéraire global relatif au domaine de recherche $AI_1 - AI_2 - AI_4 - AI_5$

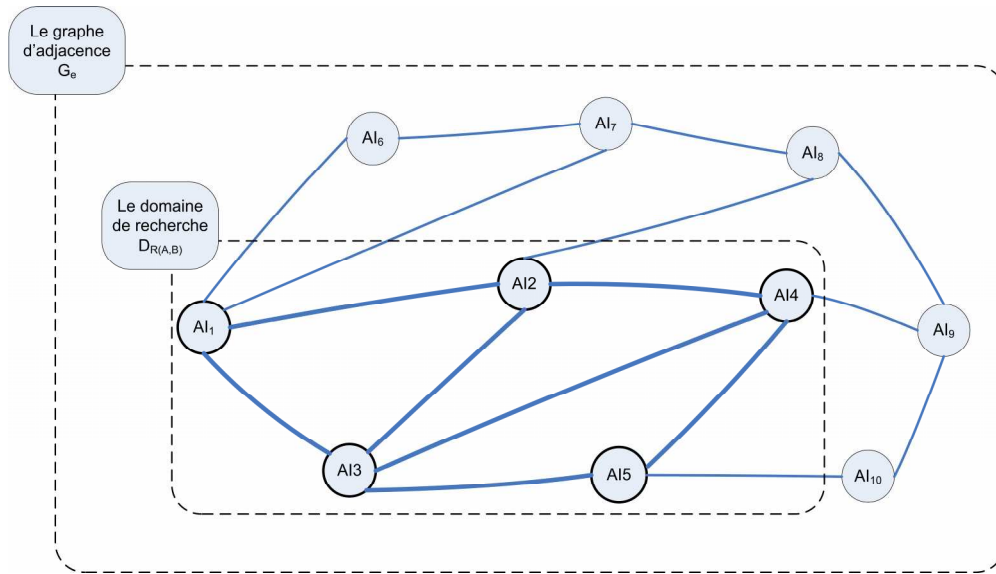


FIG. 4.6 – Domaine de recherche et graphe d'adjacence

4.2.1.2 Exemple ARCI

Deux algorithmes ont été retenus pour l'ACRI : le PCCD pour les réseaux statiques et le PCCDH qu'on a proposé pour les réseaux horaires et dynamiques. Le PCCD étant illustré au chapitre 3, nous détaillons dans cette partie les étapes d'exécution du PCCDH.

Pour valider le PCCDH nous détaillerons dans cette partie les étapes qui permettent de répondre à une requête globale $Req(S_d, S_a, t_d)$ avec S_d la station de départ, S_a celle de l'arrivée, et t_d l'instant départ de l'origine. Nous supposons que nous disposons d'un domaine de recherche $D_R(S_d, S_a)$ composé de quatre agents d'information : AI_A, AI_B, AI_C , et AI_D . On définit AI_A comme étant l'AI associé à la station de départ S_d , et AI_D comme étant l'agent associé à la station d'arrivée S_a . Nous souhaitons obtenir l'itinéraire global le plus rapide, qui arrive au plus tôt à S_a en partant de S_d à l'instant t_d .

Pour exécuter le PCCDH, l'AC devrait disposer d'un graphe d'adjacence qui lie les 4 agents entre eux, des stations d'intersection (S_1, S_2, S_3, S_4, S_5 et S_6) dont ces derniers partagent la gestion, ainsi que des deux stations S_d et S_a . En fonction de ces données, il construit le graphe partiel virtuel $G_{virt(S_d, S_a)}$ illustré par la figure 4.7. Ce graphe constitue pour l'agent courtier compositeur un guide pour l'exploration en ligne.

Pour simuler la composition en ligne réalisée par le PCCDH, chaque agent d'information doit être, par ailleurs, en mesure de répondre à une requête locale du type $req(a, b, t_a)$, avec

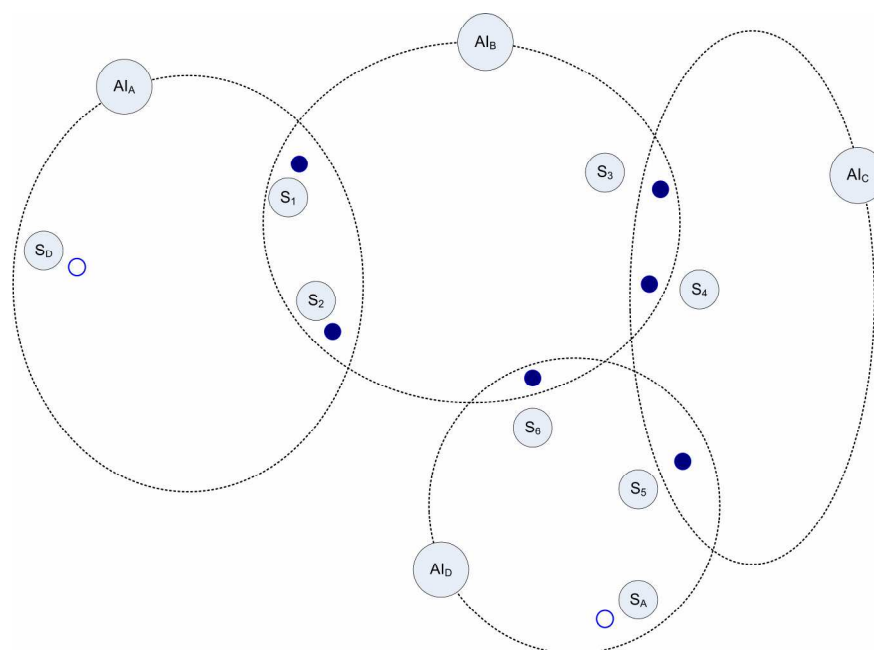


FIG. 4.7 – Exemple d'un graphe virtuel partiel

a et b des stations locales et t_a l'heure de départ de la station a . Ceci suppose que chaque AI communique avec un SIAD qui est capable de fournir l'itinéraire local qui permet, en partant de a à t_a d'arriver au plus tôt à b .

Pour simuler les SIAD, nous nous proposons alors d'associer à chaque couple de stations d'intersections, un ensemble de solutions : chemins qui permettent d'aller de a vers b , avec différents horaires de départs possibles. Aussi, quand a et b appartiennent les deux à différents SIAD, nous associons à ce couple des solutions qui proviennent aussi de ces différents SIAD.

Ceci constitue pour chaque SIAD une base de solutions : des réponses possibles aux différentes requêtes éventuelles qui pourraient être lancées par l'agent courtier compositeur. Mais ceci nous ramène à voir le réseau autrement : nous pouvons en effet associer à chaque SIAD une ligne de transport (bus, ou métro), avec des courses possibles dans les deux sens. Ces lignes présentent les mêmes pôles d'échange que les SIAD. Ainsi chaque SIAD est dans ce cas responsable d'une ligne ; il est capable de répondre à une requête $req(a, b, t_a)$ si a et b sont deux stations gérées par la même ligne.

Le graphe $G_{vint(S_d, S_a)}$ est dans ce cas un modèle partiel d'un réseau dynamique illustré par la figure 4.8, composé de quatre ligne de bus : A, B, C, et D ; associés respectivement aux agents AI_A , AI_B , AI_C , et AI_D . Pour les données horaires, nous avons associé les horaires de

passage aux arrêts des tableaux Tab 4.1, et Tab 4.2 , aux différentes lignes de bus étudiées : A, B, C et D.

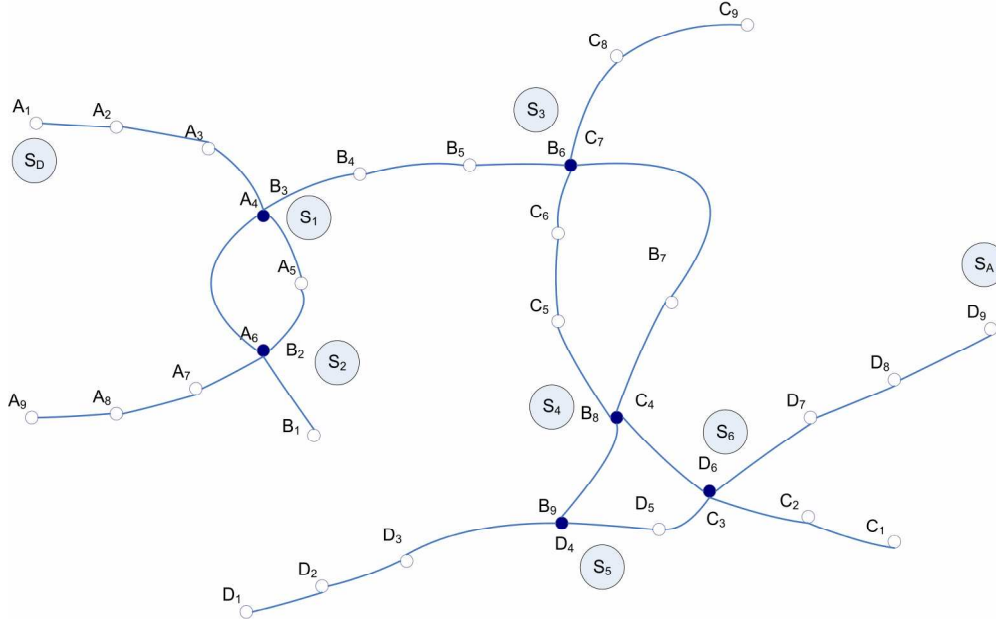


FIG. 4.8 – Exemple d'un réseau de transport avec 4 lignes de bus

Ayant ce graphe virtuel, l'agent courtier commence l'exploration en ligne des itinéraires en exécutant le PCCDH. Nous détaillons dans l'annexe, pour une requête globale $Req(S_d, S_a, t_d)$ avec $S_d = A_1$, $S_a = D_9$, et $t_d = 08 : 25$, les étapes de l'exécution de cet algorithme et l'ensemble des requêtes locales envoyées aux différents agents d'information.

A partir de S_d , station de AI_A , l'AC envoie les requêtes locales $req(S_d, S_1, 8 : 25)$ et $req(S_d, S_2, 08 : 25)$ à l'agent AI_A . S_1 et S_2 étant les stations accessibles à partir de S_d . Or pour retourner des réponses, ces requêtes doivent être adressées à l'agent d'information avec les noms locaux des stations : $req(A_1, A_4, 08 : 25)$ et $req(A_1, A_6, 08 : 25)$.

Ces requêtes vont permettre à l'AC de mettre à jour sur le graphe virtuel, les heures d'arrivée au plutôt à S_1 et S_2 , qui vont respectivement avoir les valeurs de $08 : 37$ et $08 : 47$.

La station ayant l'heure d'arrivée la plus avantageuse étant S_1 , l'AC continue l'exploration à partir de ce noeud. Or S_1 est noeud partagé entre AI_A et AI_B . Les noeuds accessibles à S_1 via AI_B sont S_2 , S_6 , S_4 et S_3 via AI_B . Seul le noeud S_2 reste accessible via AI_A . On ne considère pas le noeud S_d , puisque'il fait partie des prédécesseurs de S_1 (voir Fig 4.7) .

Au bout de 30 requêtes, l'AC détermine le chemin le plus rapide. Il s'agit dans cet exemple

A1	A2	A3	A4	A5	A6	A7	A8	A9
07 :00	07 :03	07 :05	07 :08	07 :12	07 :18	07 :24	07 :29	07 :34
07 :11	07 :14	07 :16	07 :19	07 :23	07 :29	07 :34	07 :39	07 :44
07 :12	07 :25	07 :27	07 :30	07 :34	07 :40	07 :45	07 :50	07 :55
07 :33	07 :37	07 :40	07 :42	07 :46	07 :52	07 :57	08 :02	08 :07
07 :44	07 :48	07 :50	07 :52	07 :57	08 :03	08 :08	08 :13	08 :18
07 :55	07 :59	08 :01	08 :04	08 :08	08 :14	08 :18	08 :24	08 :29
08 :06	08 :10	08 :12	08 :15	08 :19	08 :25	08 :30	08 :35	08 :40
08 :17	08 :21	08 :23	08 :26	08 :30	08 :36	08 :41	08 :46	08 :51
08 :28	08 :32	08 :34	08 :37	08 :41	08 :47	08 :52	08 :57	09 :02
08 :39	08 :42	08 :44	08 :47	08 :51	08 :57	09 :01	09 :06	09 :11
08 :50	08 :53	08 :55	08 :58	09 :02	09 :08	09 :12	09 :17	09 :22
09 :01	09 :04	09 :06	09 :09	09 :13	09 :19	09 :23	09 :28	09 :33
09 :22	09 :25	09 :27	09 :30	09 :34	09 :40	09 :43	09 :49	09 :54
09 :40	09 :43	09 :45	09 :48	09 :52	09 :58	10 :02	10 :07	10 :12
10 :00	10 :03	10 :05	10 :08	10 :12	10 :18	10 :22	10 :27	10 :32

TAB. 4.1 – Horaires de passages du bus A, de A1 vers A9

A9	A8	A7	A6	A5	A4	A3	A2	A1
07 :00	07 :03	07 :05	07 :08	07 :12	07 :18	07 :24	07 :29	07 :34
07 :11	07 :14	07 :16	07 :19	07 :23	07 :29	07 :34	07 :39	07 :44
07 :12	07 :25	07 :27	07 :30	07 :34	07 :40	07 :45	07 :50	07 :55
07 :33	07 :37	07 :40	07 :42	07 :46	07 :52	07 :57	08 :02	08 :07
07 :44	07 :48	07 :50	07 :52	07 :57	08 :03	08 :08	08 :13	08 :18
07 :55	07 :59	08 :01	08 :04	08 :08	08 :14	08 :18	08 :24	08 :29
08 :06	08 :10	08 :12	08 :15	08 :19	08 :25	08 :30	08 :35	08 :40
08 :17	08 :21	08 :23	08 :26	08 :30	08 :36	08 :41	08 :46	08 :51
08 :28	08 :32	08 :34	08 :37	08 :41	08 :47	08 :52	08 :57	09 :02
08 :39	08 :42	08 :44	08 :47	08 :51	08 :57	09 :01	09 :06	09 :11
08 :50	08 :53	08 :55	08 :58	09 :02	09 :08	09 :12	09 :17	09 :22
09 :01	09 :04	09 :06	09 :09	09 :13	09 :19	09 :23	09 :28	09 :33
09 :22	09 :25	09 :27	09 :30	09 :34	09 :40	09 :43	09 :49	09 :54
09 :40	09 :43	09 :45	09 :48	09 :52	09 :58	10 :02	10 :07	10 :12
10 :00	10 :03	10 :05	10 :08	10 :12	10 :18	10 :22	10 :27	10 :32

TAB. 4.2 – Horaires de passages du bus A, de A9 vers A1

du chemin illustré par tableau suivant (Tab 4.3) :

tronçon d'itinéraires	(S_d, S_1)	(S_1, S_4)	(S_4, S_6)	(S_6, S_a)
Heure d'arrivée retenue	08 :37	09 :06	09 :12	09 :33
AI origine de l'offre	AI_A	AI_B	AI_C	AI_D
Détails de l'offre	$A_1 - A_2 - A_3 - A_4$	$B_3 - B_4 - B_5 - B_6 - B_7 - B_8$	$C_4 - C_3$	$D_6 - D_7 - D_8 - D_9$

TAB. 4.3 – Chemin le plus rapide inter-opérateurs

4.2.2 Discussion

Pour simuler le PCCDH, nous avons associé un agent d'information à chaque ligne de transport. En effet un agent d'information invoqué par l'agent courtier, peut être associé soit au SIAD d'un opérateur multimodal, soit à une fiche horaire d'un simple mode de transport. De ce fait, il devient nécessaire d'adopter une organisation multi-agents qui tient compte de cette différence d'échelles entre les différents systèmes d'information d'aide à la décision.

Par ailleurs, il devient aussi possible d'inclure dans le domaine de recherche des agents d'information associés à des systèmes d'information qui renseignent sur le déplacement en voiture privée, en assimilant les parkings proches des points d'accès au transport public, à des stations d'intersection.

L'algorithme PCCDH présenté permet à l'AC de rechercher et de composer en ligne un itinéraire global qui répond à une requête $Req(S_d, S_a, t_d)$ sur un territoire multi-opérateurs et multimodal, en récupérant des offres locales résultats des requêtes $req(a, b, t_a)$, de chaque AI associé à un opérateur ou à un mode de transport.

Ce procédé de recherche et de composition d'itinéraire permet aussi de récupérer l'itinéraire global le plus rapide qui va de S_d à S_a , arrive à S_a à t_a en partant au plus tard de la station de départ S_d . Il suffit d'envoyer les requêtes locales $req(a, b, t_b)$ qui retournent les offres des courses les plus rapides qui partent au plus tard de a pour arriver à b , à une heure inférieure ou égale à t_b .

Le PCCDH peut tenir compte d'un temps de transit dans les gares ou les stations. Ce temps permet à un usager de se déplacer d'un quai vers un autre, pour changer de mode dans une station correspondant à un sommet d'intersection. Certes, il est difficile de l'estimer, mais il peut s'avérer nécessaire d'inclure cette marge de sécurité dans les requêtes locales,

pour récupérer des offres valables qui laissent le temps au voyageur de se déplacer dans les pôles d'échange. Une connaissance du terrain permettrait d'affiner ces temps de transit.

Dans ce cas, les requêtes locales émises par l'AC contiennent un sommet de provenance, un sommet de destination, et un horaire de départ au plus tôt $t + \Delta t$, qui correspond à l'heure d'arrivée effective au quai de départ en tenant compte du temps de transit Δt . De manière similaire, lorsque la contrainte temporelle est une heure d'arrivée au plus tard, la requête contient un horaire $t - \Delta t$. Le temps de transit Δt est paramétré, avant le lancement du PCCDH, soit par l'utilisateur, soit par le système.

4.3 Plate-forme multi-agents pour le SICM

L'organisation multi-agent proposée pour le SICM est l'organisation Interfaces-Médiateurs-adaptateurs (voir Fig 4.9) présentée et détaillée au chapitre 2. Nous proposons dans cette partie les scénarios de déploiement de cette organisation sur un réseau informatique, et les technologies utilisées pour son implémentation.

4.3.1 Vers une organisation holonique

" Il y a environ 30 ans, l'écrivain et philosophe hongrois Arthur Koestler qui étudiait les organismes vivants et les organisations sociales a mis en évidence le fait que dans la vie réelle, une entité, pouvant être un tout ou une partie, ne peut être considérée comme absolue. Il a alors introduit le concept de Holon, concaténation de holos, signifiant le tout et du suffixe en suggérant une partie comme dans 'proton' ". Ainsi, [Mandiau et al.02b] ont introduit le concept holon. Une architecture multi agents holonique, est basée sur ce concept : un agent est composé d'agents mais il est aussi composant d'agents. On peut alors considérer le SICM comme une organisation multi-agents holonique, ou chaque agent d'information qui intervient peut être lui même un SICM ou associé à un SICM.

Un SICM est en effet autonome : il peut répondre seul, à une requête d'itinéraire. Il peut aussi coopérer avec d'autres SICM, et être considéré comme un AI parmi d'autres dans un SICM de niveau plus élevé. On peut alors imaginer qu'un SICM Européen, est composé de plusieurs SICM nationaux. Un SICM national est composé de plusieurs SICM départementaux. Un SICM départemental ou régional est composé de plusieurs SICM associés aux

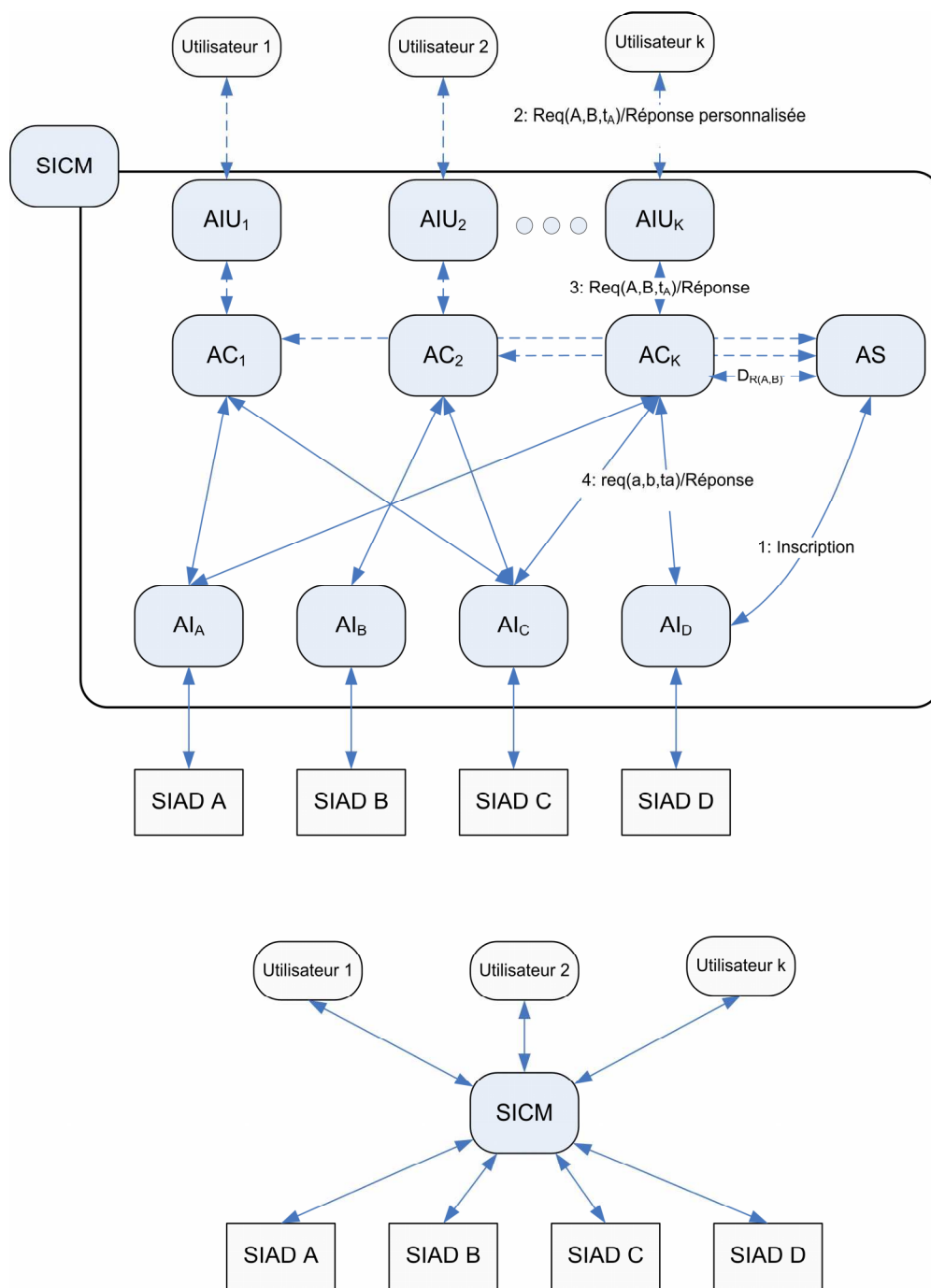


FIG. 4.9 – Organisation multi-agents du SICM

autorités organisatrices du transport. Et on peut descendre dans cette échelle pour définir des SIAD monomodaux.

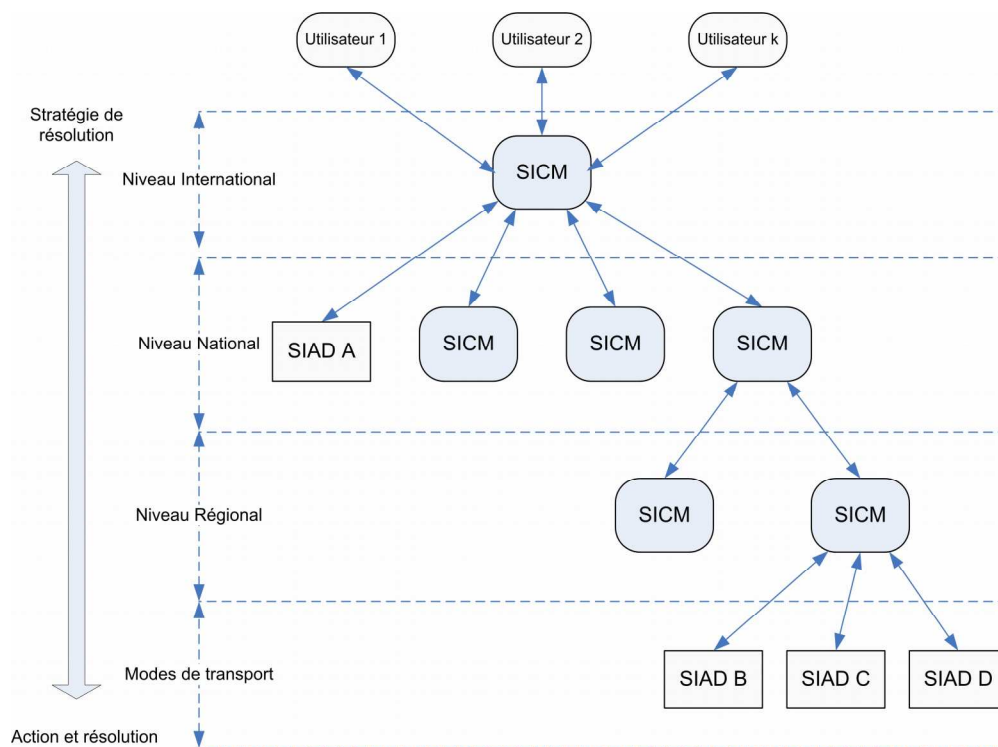


FIG. 4.10 – Organisation holonique

4.3.2 Scénarios de déploiement

Avant de procéder à l'implémentation, nous avons jugé nécessaire de proposer des scénarios de déploiements des différents agents sur un réseau informatique et notamment sur internet. En effet, le SICM est conçu pour répondre à une ou plusieurs requêtes d'utilisateurs, et ce en accédant aux différents SIAD ou aux différents SICM existants sur le réseau internet.

Les AI assurent la communication via internet entre le système et les différents SIAD, mais également entre le système et les autres SICM existants dans cet environnement. Nous avons exposé dans le chapitre 1, différentes technologies d'intégration de l'information ; la plus utilisée est celle des services web car elle se base sur un échange standardisé de flux XML. Cette technologie peut être adoptée par les SIAD pour qu'il puissent être interrogés à distance par les différents AI.

Par ailleurs pour qu'un agent d'information d'un SICM puisse interroger un autre SICM, la même technologie peut être adaptée pour permettre à deux systèmes multi-agents hétérogènes de communiquer. Des travaux ont été récemment menés par la FIPA (Foundation

of Intelligent Physical Agents) pour proposer des mécanismes de communications entre différentes plate-formes hétérogènes.

Enfin dans le système multi-agent lui même (SICM), les agents communiquent en utilisant un langage qui leur est propre, tel que l'ACL (Agent Communication Language) ou le KQML (Knowledge Query and Manipulation Language (§2.2.4)).

Le SICM étant multi-utilisateurs, il doit être capable de traiter plusieurs requêtes d'itinéraires simultanément. Or chaque requête nécessite une composition en ligne, assurée par un agent courtier, de différentes offres locales.

Pour optimiser le système global nous proposons dans l'organisation adoptée de distribuer le traitement et le calcul sur plusieurs agents courtier compositeurs. Ainsi nous avons associé, à chaque utilisateur, un agent interface utilisateur (AIU) et un agent courtier (AC) qui ne traitent que les requêtes relatives à ce dernier.

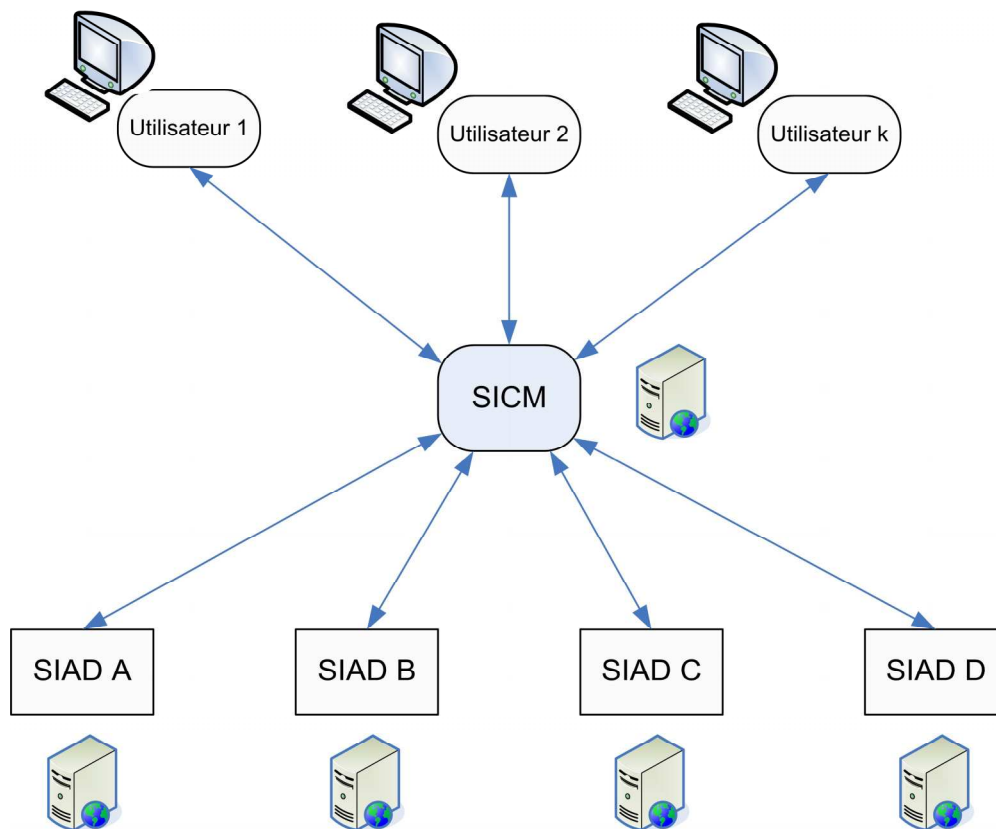


FIG. 4.11 – SICM avec un traitement centralisé

Mais si les AC associés aux différents utilisateurs font partie d'un SICM hébergé sur un seul serveur, ceci ne peut améliorer le temps du traitement global. En effet, pour distri-

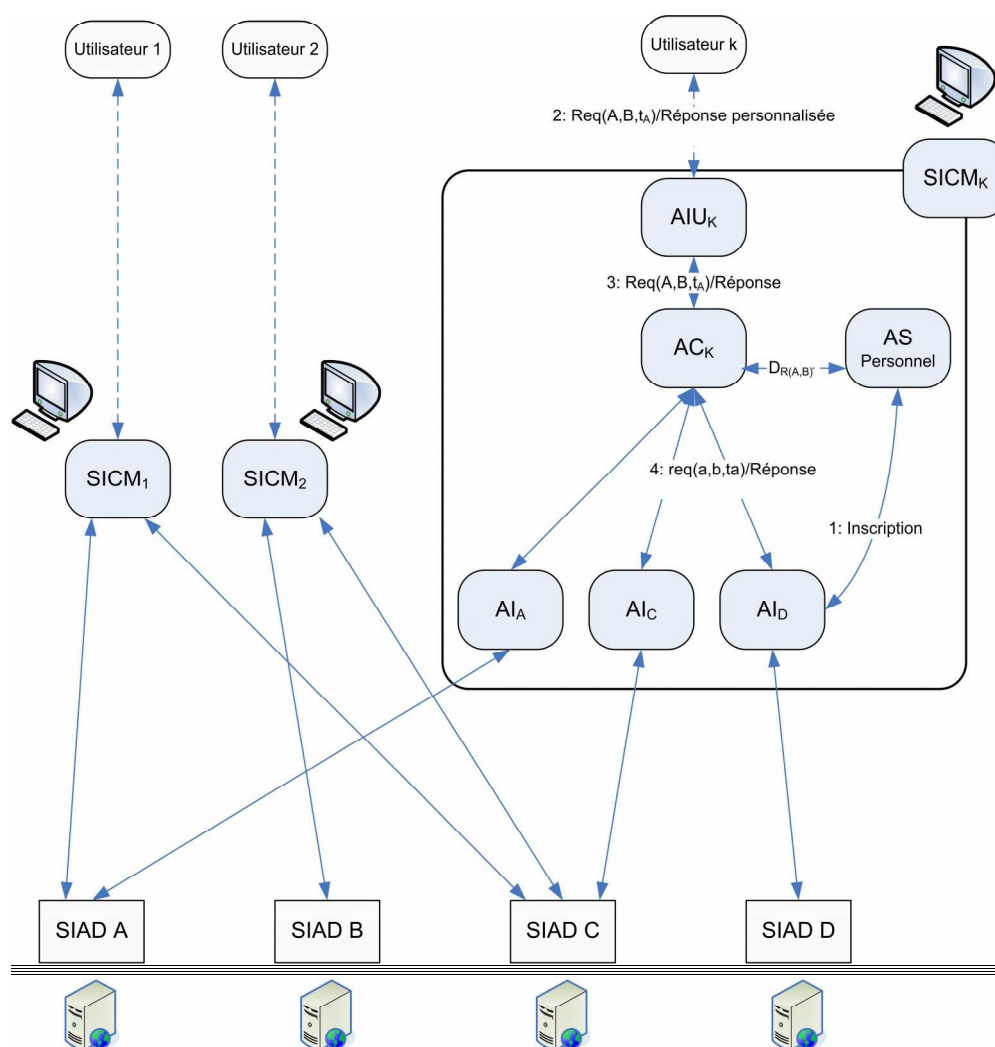


FIG. 4.12 – SICM personnalisés avec un traitement distribué

buer le traitement, il faut que les AC qui assurent la composition des différentes requêtes, soient déployés sur différentes machines. Nous proposons alors pour le déploiement les deux approches suivantes :

- La première approche consiste à développer le SICM en tant qu'application distribuée sur un réseau local. Les AC seront déployés sur différentes machines. Et le système sera ainsi accessible à distance par les différents utilisateurs.
- La deuxième approche consiste à associer un SICM à chaque utilisateur. Ainsi chaque usager disposerait sur sa machine de l'ensemble des agents du SICM : le traitement associé à ses propres requêtes s'effectue sur sa propre machine (voir Fig 4.12).

Il s'agit là d'un système personnel de recherche et de composition d'itinéraires. Ce

système peut encore être personnalisé, si l'utilisateur choisit dès son installation, l'ensemble des AI qu'il voudrait impliquer dans ses recherches. Chaque usager définit ainsi, en fonction de son besoin personnel de mobilité, son propre annuaire de transport.

4.3.3 Implémentation

Pour développer le système d'information coopératif de mobilité, nous avons commencé par implémenter les agents d'information qui vont permettre au système d'accéder à distance aux différents SIAD existants pour récupérer les différentes offres d'itinéraires locaux.

Nous avons ensuite implémenté l'algorithme de sélection du domaine de recherche (ASDR) et celui de la recherche et de la composition en ligne (ACRI) détaillés précédemment. Une fois les algorithmes validés, nous avons associé ces compétences, respectivement, à l'agent annuaire sélectionneur (AS) et à l'agent courtier (AC). Enfin nous avons implémenté l'ensemble des agents du SICM afin de tester le scénario global de communication.

4.3.3.1 Implémentation des agents d'informations

La première compétence d'un agent d'information (AI), appelé aussi dans la littérature adaptateur (wrapper) ou robosoft, est l'extraction automatisée de l'information (voir §2.3.1.1). L'extraction de l'information d'un SIAD suppose un minimum de coopération de l'opérateur de transport qui lui est associé. En effet ce dernier devrait mettre à disposition du développeur de l'AI, le protocole de communication qui précise : la syntaxe d'une requête locale $req(a, b, t_a)$, et le format d'une réponse envoyée.

A défaut de cette coopération, il reste cependant possible d'accéder à cette information sur les offres locales, à partir des sites web des différents opérateurs. Le robosoft se comporte alors comme un internaute : il accède au site web, analyse la page web HTML, remplit automatiquement les formulaires avec les données nécessaires à sa requête (station de départ, station d'arrivée et critère de sélection), et récupère ensuite les réponses proposées. Des logiciels tels que "Automate de Unisyn" ¹, ou "visual web task" ² permettent de construire ces type d'agents [Kamoun et al.03].

Les inconvénients de cette démarche, d'un point de vue technique, sont :

¹www.unisyn.com/automate/

²www.sharewareriver.com/products/5428.htm

- Devoir développer un AI pour chaque interface web. Ainsi si cette interface change, l'AI ne peut plus assurer l'extraction des données.
- Un temps de réponse très lent : il s'agit en effet d'analyser plusieurs pages web avant de récupérer les solutions requises.
- L'absence d'un accord avec le fournisseur du service ne favorise pas l'évolution du système.

Par ailleurs, le développement de la normalisation relative à l'information de transport (normes TRIDENT et SIRI, voir §1.5.2.3), s'est accompagnée en Europe d'une standardisation des services web qui permettent d'accéder à distance aux SIAD des différents opérateurs. Nous optons alors pour cette technologie pour développer les agents d'information (AI).

Pour ce faire, nous commençons par développer les services web associés au différents SIAD adoptés dans l'exemple de validation de l'ACRI : SIAD A, SIAD B, SIAD C et SIAD D. On définit un service web transport par deux méthodes accessibles à distance :

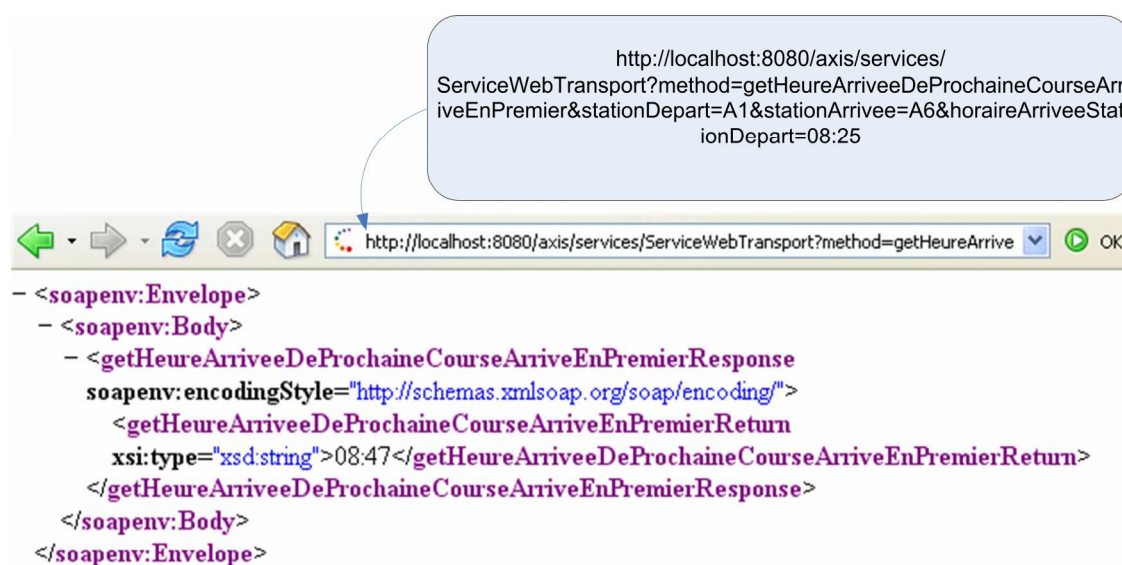


FIG. 4.13 – Réponse XML à une requête $req(A1, A6, 08 : 25)$

- une méthode qui permet de répondre à une requête locale $req(a, b, t_a)$ par l'heure d'arrivée au plus tôt de l'itinéraire qui permet d'aller de a vers b en partant de a à l'instant t_a . Il s'agit de la méthode "getHeureArriveeDeProchaineCourseArrive". La figure 4.13 détaille la réponse reçue en format XML par l'AI_A, suite à une requête $req(A1, A6, 08 : 25)$ envoyée SIAD A.

- une méthode qui permet de répondre à une requête locale $req(a, b, t_a)$ par le détail de l'itinéraire local qui permet d'aller de a vers b en partant de a à l'instant t_a . Il s'agit de la méthode "getDetailsProchaineCourseArriveEnPremier". La figure 4.14 détaille, de même, la réponse reçue en format XML, par un AI, suite à une requête $req(A1, A6, 08 : 25)$ envoyée au SIAD A.

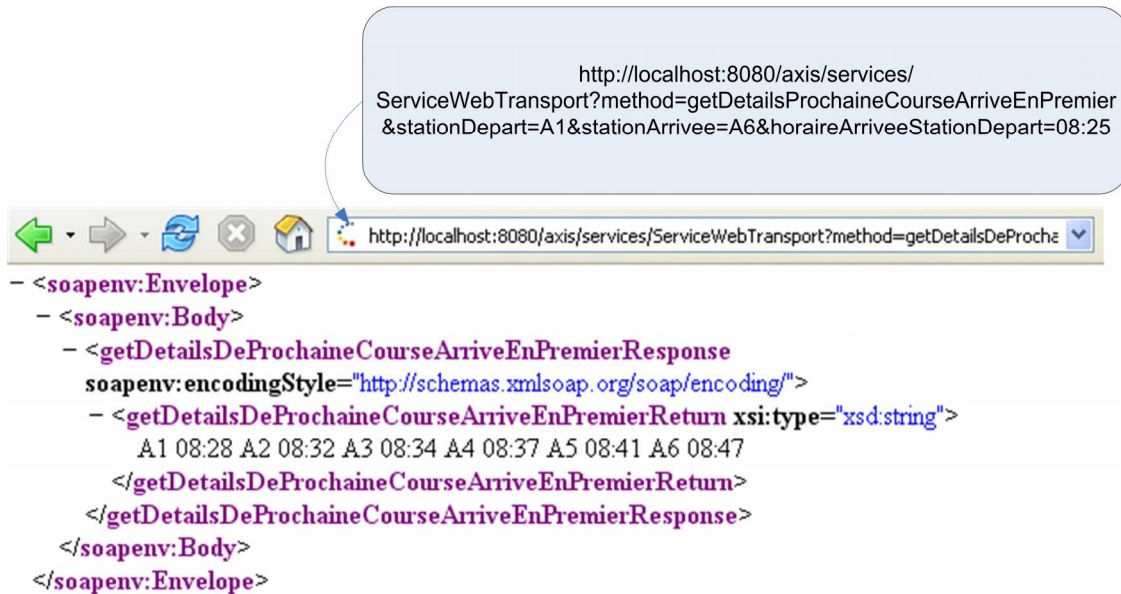
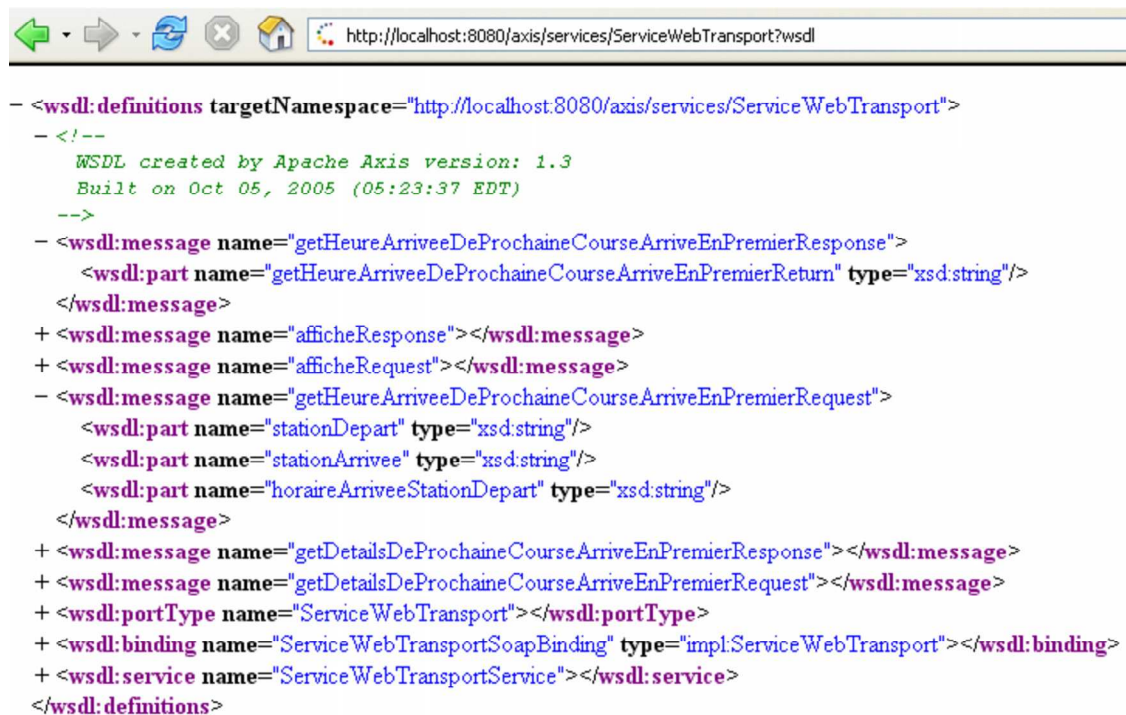


FIG. 4.14 – Réponse XML détaillée à une requête $req(A1, A6, 08 : 25)$

La figure 4.15 illustre une description XML des méthodes publiés : description du protocole d'échange entre un SIAD et un AI. Cette description permet de générer la classe dite client du service web, et qui permet à l'AI d'accéder à distance à ce service. La description est écrite en WSDL (Web Service Description Language) : langage de description du service web.

4.3.3.2 Implémentation du SICM

Une plate-forme multi-agents est un ensemble de bibliothèques de programmes qui aident à la conception, au développement et à la simulation d'un système multi-agents . Ainsi pour développer un système multi-agent, il devient pratique d'utiliser de telles plate-formes. Or, dans la littérature il existe une panoplie d'outils réalisés par des équipes de recherche et qui restent adaptés à leurs organisation multi-agents, leurs modèles d'agents et leurs applications [Mandiau et al.02b]. Nous citons :



```

- <wsdl:definitions targetNamespace="http://localhost:8080/axis/services/ServiceWebTransport">
- <!--
  WSDL created by Apache Axis version: 1.3
  Built on Oct 05, 2005 (05:23:37 EDT)
-->
- <wsdl:message name="getHeureArriveeDeProchaineCourseArriveEnPremierResponse">
  <wsdl:part name="getHeureArriveeDeProchaineCourseArriveEnPremierReturn" type="xsd:string"/>
</wsdl:message>
+ <wsdl:message name="afficheResponse"></wsdl:message>
+ <wsdl:message name="afficheRequest"></wsdl:message>
- <wsdl:message name="getHeureArriveeDeProchaineCourseArriveEnPremierRequest">
  <wsdl:part name="stationDepart" type="xsd:string"/>
  <wsdl:part name="stationArrivee" type="xsd:string"/>
  <wsdl:part name="horaireArriveeStationDepart" type="xsd:string"/>
</wsdl:message>
+ <wsdl:message name="getDetailsDeProchaineCourseArriveEnPremierResponse"></wsdl:message>
+ <wsdl:message name="getDetailsDeProchaineCourseArriveEnPremierRequest"></wsdl:message>
+ <wsdl:portType name="ServiceWebTransport"></wsdl:portType>
+ <wsdl:binding name="ServiceWebTransportSoapBinding" type="impl:ServiceWebTransport"></wsdl:binding>
+ <wsdl:service name="ServiceWebTransportService"></wsdl:service>
</wsdl:definitions>

```

FIG. 4.15 – Description XML d'un service web

- La plate-forme Magique développée à l'université de lille 1 par l'équipe SMAC³. Implémentée en java, elle adopte une organisation hiérarchique d'agents. Elle permet d'associer facilement des services et des compétences à des agents. Elle a été utilisée par [Anli et al.04] et [Bauer et al.01] pour le développement de systèmes d'information voyageur personnalisés.
- La plate-forme Madkit implémentée aussi en java, a été développée au LIRMM⁴ (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier) sur un modèle organisationnel appelé Agents/Groupe/Rôle [Ferber et al.98]. Ce modèle associe à chaque agent un rôle spécifique au sein d'un groupe.
- La plate-forme JADE⁵ (Java Agent DEvelopment Framework), développée au départ par Telecom Italia a été soutenue ensuite par un consortium d'entreprises. Jade est définie comme un intergiciel orienté agents. Elle est souvent utilisée pour la réalisation des systèmes d'information orientés agents [Berger et al.02].

³<http://sma.lip6.fr/equipes/smaclifl.htm>

⁴<http://www.lirmm.fr>

⁵<http://jade.tilab.com/>

Face à cette diversité d'organisations, de modèles et d'applications, il devient difficile de distinguer les avantages ou les inconvénients d'une plate-forme par rapport à une autre. Le choix est généralement orienté par le degré d'adéquation entre les spécifications du système multi-agents à réaliser et les fonctionnalités offertes par la plate-forme.

Pour implémenter le SICM nous avons opté pour l'utilisation de la plate-forme JADE. Ce choix a été motivé par les raisons suivantes :

- JADE est fondée sur les spécifications FIPA. Ces spécifications ont été élaborées afin de faciliter la communication entre différentes plate-formes multi-agent hétérogènes. L'adopter permet alors à un SICM de communiquer avec d'autres SICM hétérogènes.
- JADE peut gérer un système multi-agents dont les agents sont répartis sur un réseau informatique. Installée sur différentes machines, elle permet en effet à des agents répartis de communiquer d'une manière transparente à travers le réseau. Ceci permet de déployer les agents courtiers composeurs sur différentes machines pour distribuer le traitement.
- Une extension de cette plate-forme appelée (WSIG : Web Service Integration Gateway) permet aussi à des agents de JADE d'accéder à distance à des services web. Elle permet également à des agents d'être invoqués à distance en tant que services web.

Pour toutes ces raisons nous avons implémenté les agents du SICM sur JADE. La figure 4.16 illustre les communication inter-agents réalisées en ACL (Agent Communication Language) pour répondre à une requête globale d'itinéraire.

4.4 Conclusion

Nous avons présenté dans ce chapitre des exemples de simulation des différents algorithmes adoptés par les agents du SICM. Nous avons présenté différentes stratégies de déploiement du système d'information coopératif de mobilité. Pour tester le fonctionnement global du système, nous avons implémenté les agents sur la plate-forme JADE.

Le développement réalisé constitue un prototype qui démontre la faisabilité d'un tel système d'information coopératif se basant sur une composition dynamique de services web. Par ailleurs, les données utilisées restent limitées, et ne permettent pas de tester les performances globales du système en cas de montée de charge (nombre de requêtes très élevées), ou dans

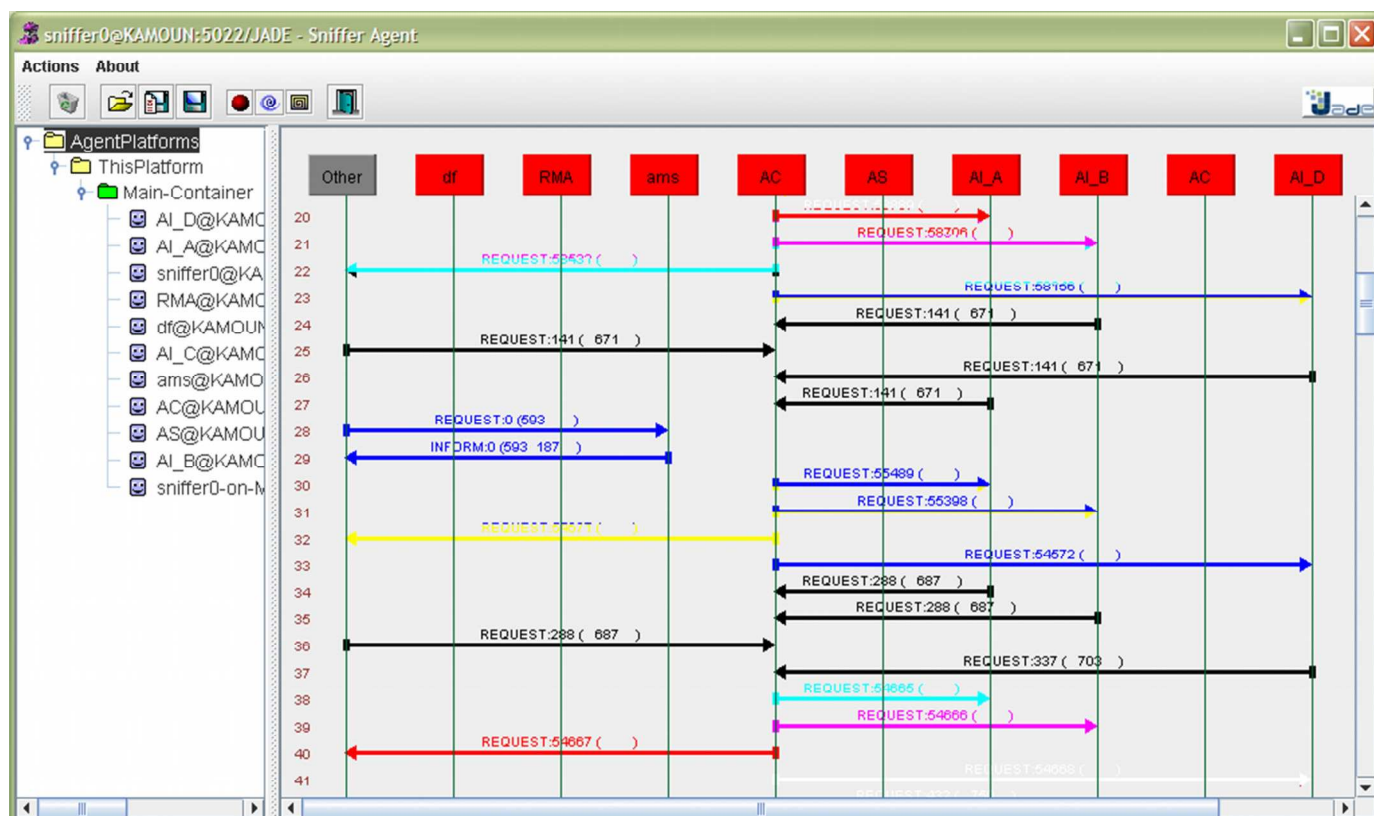


FIG. 4.16 – Exemple de communication des agents du SICM sur Jade

le cas d'un nombre de stations d'intersection très élevé.

Pour vérifier les performances du système dans un contexte réel, nous procédons actuellement à son intégration dans Le projet VIATIC.Mobilité⁶ du pôle de compétitivité I-Trans.

⁶<http://viatic.inrets.fr/>

Conclusion et perspectives

Dans ce travail de recherche nous avons présenté une nouvelle approche orientée agents pour l'intégration dynamique et en ligne de l'information voyageur à partir des systèmes d'information de différents exploitants de transport. Nous avons, en effet proposé un système d'information orienté agents, qui permet d'interroger les systèmes d'information existants, et qui, pour fournir une information multimodale et multi-opérateurs, va compiler les résultats de recherche de différents calculateurs d'itinéraires.

Ce système d'information orienté agents que nous appelons dans ce mémoire, système d'information coopératif de mobilité ou SICM, est composé de plusieurs agents d'informations qui communiquent et coopèrent entre eux pour répondre à une requête d'un itinéraire global. Les agents sélectionnent les systèmes d'information à interroger pour définir un domaine de recherche. Ils recherchent ensuite, sur ce domaine, les meilleures combinaisons d'offres locales pour composer l'itinéraire global. Le SICM est une organisation de systèmes d'informations coopératifs : des calculateurs d'itinéraires communicants, qui en répondant chacun à un problème d'optimisation sur le réseau de transport qu'il gère, vont répondre ensemble à un problème d'optimisation global sur un réseaux de transport étendu, multimodal et multi-opérateurs.

Vu la complexité du système à réaliser nous avons procédé à une conception en trois étapes :

- Nous avons commencé par chercher dans la littérature, les organisations de systèmes d'information orientés agents, qui s'adaptent le mieux à notre problématique de recherche et de composition de l'information. L'organisation Interface-Médiateurs-Adaptateurs étant adoptée, nous avons détaillé les comportements et la communication des différents agents qui la composent dans le SICM.
- Nous avons ensuite proposé des algorithmes distribués pour la recherche et la compo-

sition d'itinéraires. Ces algorithmes sont des compétences associées aux agents précédemment définis. Nous avons adopté et conçu ces algorithmes en tenant compte des caractéristiques des réseaux de transport : statiques ou dynamiques (dépendants du temps).

- Enfin, nous avons validé les algorithmes et implémenté le système multi-agents. Nous avons aussi proposé des scénarii de déploiement du SICM qui permettent de distribuer le traitement des requêtes, et des technologies d'implémentation qui prennent en considération les standards actuels (XML, web service, EAI).

Ce travail de recherche a fait l'objet de quatre publications dans différents congrès et de deux revues. Nous citons : WESEAS 2004 [Kamoun et al.04], REE [Ben khaled et al.05], Doctoriales INRETS 2005 [Kamoun et al.05b], MHOSI 2005 [Kamoun et al.05c], IMACS 2005 [Kamoun et al.05a] et IEEE SMC'05 [Kamoun et al.05d]. Dans ces travaux nous proposons : et des organisations de systèmes multi-agents pour la recherche et la composition d'itinéraire, et les algorithmes recherche et de composition qui leur sont associés.

Par ailleurs un brevet a été déposé par l'INRETS⁷, Fin Août 2006, avec comme co-inventeur l'Ecole Centrale de Lille⁸, dans lequel nous proposons un procédé de recherche et de composition d'itinéraires multimodal et multi-opérateurs en ligne adapté aux réseaux dynamiques et horaires. Nous proposons dans ce brevet différents scénarios de déploiements, et différentes applications du procédé.

Cette approche remet l'utilisateur du transport au centre de la problématique de mobilité. En effet, avec le système d'information proposé, nous passons d'une information multimodale diffusée par l'exploitant, à une information multimodale recherchée et composée pour le voyageur. Cette composition d'itinéraires est alors réalisée selon les critères de ce dernier, et met en concurrence les offres locales des différents fournisseurs de services. De plus, avec un tel système, l'utilisateur disposerait d'une continuité servicielle de l'information voyageur : Il peut disposer d'une information multimodale sur un territoire étendu, sans se soucier des cloisonnements juridiques, institutionnels et même technologiques qui séparent les différents opérateurs de transport.

⁷<http://www.inrets.fr/>

⁸<http://www.ec-lille.fr/>

Les perspectives restent nombreuses en termes d'applications, et vont du calculateur d'itinéraire personnel et personnalisé, au calculateur d'itinéraires Européen ou international. Mais avant cela nous nous proposons d'étudier les performances globales du système afin d'optimiser le temps de réponse global. Nous comptons par ailleurs intégrer ce travail dans le projet VIATIC.MOBILITE⁹ du pôle de compétitivité I-Trans. Ce projet vise à fournir une information voyageur multimodale et multi-opérateurs à domicile, aux pôles d'échanges et au bord des TER en déplacement. Nous adapterons nos algorithmes aux spécifications du VIATIC.MOBILITE, et nous profiterons de la disponibilité des gisements de données transport pour tester les performances du SICM.

Dans le cadre de ce projet, nous étudierons également avec nos collègues du LAMIH-
footnote<http://www.univ-valenciennes.fr/LAMIH/>, les associations possibles entre les systèmes multi-agents de personnalisation de l'information, et notre système de recherche et de composition. Enfin, nous essayerons d'appliquer cette démarche composition dynamique d'itinéraires -dans le domaine du transport de marchandises- pour composer les offres de différents transporteurs sur un territoire étendu.

⁹<http://viatic.inrets.fr/>

Glossaire des abréviations

Agents

AI : Agent d'Information

AIU : Agent Interface Utilisateur

AC : Agent Courtier

AS : Agent Annuaire Sélectionneur

Algorithmes

ACRI : Algorithme de Recherche et de Composition d'itinéraires

ASDR : Algorithme de Sélection du Domaine de Recherche

PCCD : Algorithme du Plus Court Chemin Distribué

PCCDH : Algorithme du Plus Court Chemin Distribué et Horaire

Systèmes

SIM : Système d'Information Multimodale

SAE : Système d'aide à l'Exploitation

SIAD : Système d'Information d'Aide au Déplacement

SICM : Système d'Information Coopératif de Mobilité

Bibliographie

- [Adam00] E. Adam. *Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise : application aux systèmes administratifs complexes*. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, Septembre 2000.
- [Afnor97] AFNOR. *Dictionnaire de l'informatique : le vocabulaire normalisé*. ISO AFNOR, 1997.
- [Ahuja et al.02] R. K. Ahuja, J. B. Orlin, S. Pallottino, et M. G. Scutellà. *Dynamic Shortest Paths Minimizing Travel Times and Costs*. Rapport technique, Août 2002.
- [Anli et al.04] A. Anli, C. Petit-Roze, et F. Grislin-Le Strugeon. User-based decision support system. Novembre 2004.
- [Anli06] A. Anli. *Méthodologie de développement des systèmes d'information personnalisés - Application à un système d'information au service des usagers des transports terrestres de personnes*. Valenciennes, Thèse de doctorat, Université de valenciennes et du Hainaut-Cambrésis, Octobre 2006.
- [Atlas04] Atlas. *Atlas des Trasports (En France sans voiture)*. Itinéraires et territoires, 2004.
- [Badeig05] F. Badeig. *Etat de l'art de la communication entre agents et plus particulièrement de l'écoute flottante*. Mémoire de D.E.A., Université Paris IX Dauphine, Mars 2005.

- [Baek et al.06] J.-W. Baek et H. Y. Yeom. A timed mobile agent planning approach for distributed information retrieval in dynamic network environments. *Information sciences*, 2006.
- [Balbo et al.04] F. Balbo et M. Zargayouna. Serveur agent d'information : application à l'information des voyageurs. In *INFORSID*, pp. 249–265. Biarritz, France, Mai 2004.
- [Bauer et al.01] B. Bauer, J. P. Muller, et J. Odell. Agent uml : A formalism for specifying multiagent interaction. In P. Ciancarini et M. Wooldridge (Eds.), *Agent-Oriented Software Engineering*, pp. 91–2003. Berlin, Germany, Springer-Verlag, 2001.
- [Bellman58] R. E. Bellman. On a routing problem. *Quarterly Applied Mathematics*, 16 : 87–90, 1958.
- [Ben khaled et al.05] I. Ben Khaled, M. Kamoun, K. Zidi, et S. Hammadi. Vers un système d'information voyageur multimodal (sim) à base de système multi-agents (sma). *REE revue de la SEE (Société de l'Electricité de l'Electronique et des Technologie de l'Information et de la Communication)*, (1) : 41–47, Janvier 2005.
- [Bennani et al.02] N. Bennani, D. Donsez, S. Lecompte, et F. Ramos. Organisation et applications des sma. pp. 321–331. Lavoisier, 2002.
- [Bergamachi et al.01] S. Bergamachi, G. Cabri, F. Guerra, L. Leonardi, M. Vincini, et F. Zmbonelli. Mobile agent for information integration. 2001.
- [Berger et al.02] M. Berger, S. Rusitschka, D. Toropov, M. Watzke, et M. S. Bologna. Porting distributed agent-middleware to small mobile devices. 2002.
- [Bousedjra05] M. Bousedjra. *Contribution à la résolution du problème du plus court chemin multiobjectif par algorithmes évolutionnistes : Application aus systèmes de transport intermodal*. Thèse de doctorat, l'Université de Franche-Comté et de l'Université de technologie de Belfort-Montbéliard, Novembre 2005.
- [Bradshaw97] J. M. Bradshaw. An introduction to software agents. In J. M. Bradshaw (Ed.), *Software Agents*, pp. 3–46. AAAI Press / The MIT Press, 1997.

- [Bresciani et al.04] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, et A. Perini. Tropos : An agent-oriented software development methodology. *Journal of Autonomous Agents and Multi-Agent Systems*, 8(3) : 203–236, Mai 2004.
- [Busse et al.04] S. Busse et T. Kabish. Miweb : Mediator-based integration of web sources. 2004.
- [Camacho et al.00] D. Camacho, D. Borrajo, et J. Molina. Travelplan : A multiagent system to solve web electronic travel problems. In *Workshop on Agent-Based Recommender Systems. Fourth International Conference on Autonomous Agents*. Barcelona, Catalonia (Spain), ACM, Juin 2000.
- [Chabini et al.98] I. Chabini, A. Glenn, S. Pallottino, et M. G. Scutella. Reoptimization algorithms for minimum-time path problems in dynamic networks. 1998.
- [Chabini97] I. Chabini. Discrete dynamic shortest path problems in transportation application : Complexity and algorithms with optimal run time. *Transportation Research Records*, 1997.
- [Chauvet03] J.-M. Chauvet. De xml aux services web pour les entreprises. *Techniques de l'ingénieur*, HB2 Technologies logicielles Architecture des Systèmes(H2910), Février 2003.
- [Chen et al.98] L. Chen et k. Sycara. Webmate : A personal agent for browsing and searching. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multi Agents Systems (AAMAS'98- Minéapoli, USA, May 10-13)*. 1998.
- [Chen et al.04] Y. L. Chen et H. H. Yang. Finding k shortest paths in a time window network. *Computers & Operations Research*, 31 : 499–513, 2004.
- [Chevauss04] M. Chevauss. Entreprise application integration : Eai. *Techniques de l'ingénieur*, HB3 Technologies Logicielles Architectures des systèmes(H2915), Aout 2004.
- [Climaco et al.82] J. Climaco et E. Martins. A bicriterion shortest path algorithm. *European Journal of Operations Research*, 11 : 399–404, 1982.

- [Costa et al.03] L. H. M. K. Costa, M. D. de Amorim, et S. Fdida. Utilisant l'inondation contrôlée pour la découverte de routes dans les réseaux ad hoc. In *Gestion de Réseaux et de Services*. Brazil, Février 2003.
- [Cote et al.98] M. Cote, B. Chaib-draa, et N. Troudi. Netsa : une architecture multiagent réutilisable pour les environnements riches en informations. 1998.
- [Dam et al.03a] K. H. Dam et M. Winikoff. Comparing agent-oriented methodologies. In P. Giorgini, B. Henderson-Sellers, et M. Winikoff (Eds.), *Agent Oriented Information Systems (AOIS' 2003)*, pp. 78–93. Springer, Octobre 2003.
- [Dam et al.03b] K. H. Dam et M. Winikoff. Comparing agent-oriented methodologies. In P. Giorgini, B. Henderson-Sellers, et M. Winikoff (Eds.), *Agent-Oriented Information Systems 5th International Bi-Conference Workshop (AOIS 2003)*. Springer Verlag, 2003.
- [Danflous00] D. Danflous. *Déploiement National des Systèmes d'Information Multimodale : DELFI, l'exemple Allemand*. Rapport technique, CERTU (Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques), Aout 2000.
- [Danflous02] D. Danflous. *Déploiement National des Systèmes d'Information Multimodale*. Rapport technique, CERTU (Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques), Aout 2002.
- [DeLoach et al.01] S. A. DeLoach, M. F. Wood, et C. H. Sparkman. Multiagent system engeneering. *International Journal of Software Engineering and Knowledge Engeneering*, 11(3) : 231–258, 2001.
- [Demazeau et al.90] Y. Demazeau et J. Muller. *Decentralised artificial intelligence*. Elsevier Science Publisher, 1990.
- [Dieterich et al.97] G. V. H. Dieterich et D. Steiner. *Intelligent Agents for Personal Travel Assistance*. Rapport technique, Germany, Siemens AG, ZTI IK, 1997.

- [Dijkstra59] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Matematik*, 1 : 269–271, 1959.
- [Eppstein94] D. Eppstein. Finding the k shortest paths. In *IEEE Symposium on Foundations of Computer Science*, pp. 154–165. 1994.
- [Fan et al.99] Y. Fan et S. Gauch. Adaptive agents for information gathering from multiple distributed information sources. In *AAAI Symposium on Agents in Cyberspace*, pp. 40–46. 1999.
- [Fayech03] B. Fayech. *Régulation des réseaux de transport multimodal : Systèmes multi-agents et algorithmes évolutionnistes*. Thèse de doctorat, Ecole Centrale de Lille/ Université des Sciences et Technologies de Lille, Octobre 2003.
- [Ferber et al.98] J. Ferber et O. Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In *Third International Conference on Multi-Agent Systems (ICMAS' 98) Proceedings*. IFF Computer Society, 1998.
- [Ferber95] J. Ferber. *Les Systèmes Multi-Agents vers une intelligence collective*. InterEdition, 1995.
- [Fleury05] R. Fleury. *JAVA/XML*. Eyrolles, 2005, *Les cahiers du Programmeurs*.
- [Florez99] R. Florez. Towards a standardization of multi-agent system frameworks. *ACM Crosswords Student Magazine*, 1999.
- [Foo et al.99] H. M. Foo, H. M. Lenong, Y. Lao, et H. C. Lau. A multi-criteria, multi-modal passenger route advisory system. In *In Proc 1999 IES-CTR Int'l Symposium*. 1999.
- [Ford56] L. R. Ford. Network flow theory. *Rand corporation report*, (p-293), 1956.
- [Geen et al.97] S. Geen, L. Hurst, B. Nangle, P. Cunningham, F. Somers, et R. Evans. *Software Agents : A review*. Rapport technique, Trinity College Dublin, 1997.

- [Gendre99] P. Gendre. *Systèmes d'information multimodales*. Rapport technique, CERTU (Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques), Mars 1999.
- [Giorgini et al.03] P. Giorgini, B. Henderson, et M. Winikoff (Eds.). *Agent-Oriented Information Systems*. Springer, October 2003.
- [Greenwood et al.05] D. Greenwood, P. Buhler, et A. Reitbauer. Web service discovery and composition using the web service integration gateway. In *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*. 2005.
- [Gude catalan et al.00] S. Gude-Catalan et G. Uster. *Recherche Institutionnelle et Juridique sur L'Information Multimodale*. Rapport technique, INRETS, Octobre 2000.
- [Hammer et al.00] J. Hammer et J. Fiedler. Using mobile crawlers to search the web efficiently. *Journal of Computer and Information Science*, 1(1) : 36–58, 2000.
- [Iglesias et al.98] C. A. Iglesias, M. Garijo, J. C. Gonzalez, et J. R. Verlasco. Analysis and design of multi-agents systems using mas-commonkads. In M. P. Singh, A. Rao, et M. J. Wooldridge (Eds.), *Intelligent Agents IV*. Springer-Verlag, 1998.
- [Isorce et al.05a] P. Isorce et T. Anhnguen. *Standardisation d'un calcul d'itinéraire global - Partie 1*. Rapport technique, MT System-PREDIM, Février 2005.
- [Isorce et al.05b] P. Isorce et T. Anhnguen. *Standardisation d'un calcul d'itinéraire global - Partie 2*. Rapport technique, MT System-PREDIM, Juillet 2005.
- [Isorce et al.05c] P. Isorce et T. Anhnguen. *Standardisation d'un calcul d'itinéraire global - Partie 3*. Rapport technique, MT System-PREDIM, Février 2005.
- [Jamed et al.02] B. Jamed et al. *Pour un développement de l'information multimodale en agglomération : freins et perspectives*. Rapport technique, ITS-France, avril 2002.

- [James et al.02] B. James et S. Pean. *Pour un développement de l'information multimodale en agglomération : freins et perspectives*. Rapport technique, ATEC -ITS France, 2002.
- [Jarras et al.02] I. Jarras et B. Chaib-draa. Aperçu sur les systèmes multiagents. *les cahiers de la série scientifique (CS)*, 2002.
- [Kamoun et al.03] M. A. Kamoun, G. Uster, et S. Hammadi. *Service d'Information Multi-modale*. Rapport technique, Institut National de Recherche pour le Transport et sa Sécurité, 2003.
- [Kamoun et al.04] M. A. Kamoun et S. Hammadi. A multi-agent architecture for a transport's multi-modal information system. *WSEAS Transactions On Systems*, 5(3) : 2062, Juillet 2004.
- [Kamoun et al.05a] M. A. Kamoun, G. Uster, et S. Hammadi. An agent-based cooperative information system for multimodal transport's travelers assistance. In *IMACS 2005 Scientific Computation, Applied Mathematics and Simulation*. Paris France, Juillet 2005.
- [Kamoun et al.05b] M. A. Kamoun, G. Uster, et S. Hammadi. Optimisation des systèmes d'information coopératifs pour l'aide au déplacement multimodal. In *Rencontre avec les doctorants des laboratoires ESTAS, LEOST, LIVIC, LTN, Actes INRETS*. INRETS, 2005.
- [Kamoun et al.05c] M. A. Kamoun, G. Uster, et S. Hammadi. Système d'information coopératif à base d'agents pour la recherche d'itinéraires multimodaux. Avril 2005.
- [Kamoun et al.05d] M. A. Kamoun, G. Uster, et U. S. Hammadi The Big Island, Hawaii. An agent-based cooperative information system for multi-modal travelers route computation. In *2005 IEEE International Conference on Systems, Man, and Cybernetics*. The Big Island, Hawaii, USA, Octobre 2005.
- [Knoblock et al.97] C. A. Knoblock et J. L. Ambite. Agents for information gathering. pp. 347–373. MIT Press, 1997.

- [Lahlouhi01] A. Lahlouhi. Méthodologie de développement d'environnements de développement des sma. *Techniques et sciences informatiques*, 1(1), 2001.
- [Laichour02] H. Laichour. *Modélisation Multi-Agent et Aide à la Décision : Application à la régulation des correspondances dans les réseaux de transport urbains*. Thèse de doctorat, Université des sciences et technologies de Lille, Décembre 2002.
- [Le strugeon95] E. Le Strugeon. *Une méthodologie d'auto-adaptation d'un système multi-agents cognitifs*. Thèse de doctorat, Université de Valenciennes et du Hnaut-Cambrésis, janvier 1995.
- [Leal et al.01] G. Leal, P. Chainho, R. Evans, F. Garijo, J. Gomez, J. Pavon, P. Kearney, J. Stark, et P. Massonet. Agent oriented analysis using message/uml. In P. Ciancarini, M. Wooldridge, et G. Weiss (Eds.), *Second International Workshop on Agent Oriented Software Engineering (AOSE-2001)*, pp. 101–108. 2001.
- [Levy et al.00] A. Y. Levy et D. S. Weld. Intelligent internet systems. *Artificial Intelligence*, 118 : 1–14, 2000.
- [Lieberman97] H. Lieberman. Autonomus interface agents. In *Proceedings of Agents'97*. ACM press, 1997.
- [Lin et al.03] K. S. L. Y. F. Lin et J. Y. Chen. Using mobile agent and soaprpc in an advanced traveler information system. In *IEEE Intelligent Transportation Systems*, pp. 283–287. Octobre 2003.
- [Liu03a] W. Liu. *Conception / Réalisation d'un serveur de données pour favoriser les échanges d'informations normalisés sur les transports collectifs*. Mémoire de D.E.A., Ecole Nationale des Ponts ET Chaussées, Décembre 2003.
- [Liu03b] Y. Liu. *Conception Réalisation d'une application de référence pour la mise en ligne de bases de données sur l'offre de transports collectifs permettant des échanges de données normalisés*. Mémoire de D.E.A., Ecole Nationale des Ponts et Chaussées, Décembre 2003.

- [Maes94] P. Maes. Agents that reduce work and information overloaded. *Communication of the ACM*, 37(7) : 31–40, 1994.
- [Mandiau et al.02a] R. Mandiau et E. G. le Strugeon. Systèmes multi-agents. *Techniques de l'Ingénieur*, S7216 traité Informatique Industrielle, 2002.
- [Mandiau et al.02b] R. Mandiau, F. G. L. Strugeon, et A. Peninou. *Organisation et applications des SMA*. Lavoisier, 2002.
- [Martin et al.97] D. Martin, H. Oohama, D. Moran, et A. Cheyer. Information brokering in an agent architecture. In *In Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*. Londre, Avril 1997.
- [Martins et al.98] E. Martins, M. Pascoal, et J. Santos. *The K Shortest Paths Problem*. Rapport technique, Centre for Informatics and Systems of the University of Coimbra (CISUC), 1998.
- [Martins et al.99] E. Martins et J. D. Santos. the labelling algorithm for for the multiobjective shortest path problem. 1999.
- [Martins84] E. Martins. On a multicriteria shortest path algorithm. *European Journal of Operational Research*, 16 : 236–245, 1984.
- [Mehandjiev et al.04] N. Mehandjiev et A. Abdoessalam. Cooperative booking of time-dependent travel services. In *Proceedings of AAMAS04 Workshop on Agents for Ubiquitous Computing (UbiAgents)*. New York, USA, 2004.
- [Melliti04] T. Melliti. *Interopérabilité des service Web complexes. Application aux systèmes multi-agents*. Thèse de doctorat, Université de Paris IX Dauphine, Décembre 2004.
- [Moore59] E. F. Moore. The shortest path through a maze. In H. U. Press (Ed.), *Proceedings of the international Symposium on the theory of switching*, pp. 285–293. 1959.
- [Nodine et al.99] N. Nodine, J. Flower, et B. Perry. Active information gathering in infoleuth. In *CODAS*, pp. 15–26. 1999.

- [Nonato et al.99] M. Nonato, S. Pallottino, et B. Xnewen. *SPTL shortest path algorithms : review, new proposals and some experimental results*. Rapport technique n° TR-99-16, Juillet 1999.
- [Nwana96] H. S. Nwana. Software agents : An overview. *Knowledge Engineering Review*, 11 : 1–40, 1996.
- [Padgham et al.02] L. Padgham et M. Winikoff. Prometheus : A methodology for developing intelligent agents. In *Proceedings of the Third International Workshop on Agent Oriented Software Engineering (AAMAS' 2002)*. Bologne, Italie, 2002.
- [Pallottino et al.97] S. Pallottino et M. G. Scutellà. *Shortest Path Algorithms in Transportation models : classical and innovative aspects*. Rapport technique n° TR-97-06, 1997.
- [Panti et al.02] M. Panti, L. Penserini, L. spalazzi, et S. Tacconi. *A Multi-Agent system based on the P2P model to Information Integration*. Rapport technique, Computer Science Institute - University of Ancona, 2002.
- [Papavastrou et al.00] S. Papavastrou, G. Samaras, et E. Pitoura. Mobile agents for world wide web distributed databases. *IEEE Transactions on Knowledge and Data Engineering*, 12(5) : 802–820, 2000.
- [Park et al.05] S. Park et V. Sugumaran. Designing multi-agent systems : a framework and application. *Expert Systems with Applications*, 28 : 259–271, 2005.
- [Perreau02] C. Perreau. *Les Systèmes d'Information Multimodale*. Thèse de doctorat, Institut d'Etudes Politiques de Paris, 2002.
- [Roze et al.00] C. Roze, E. Grislin-Le Strugeon, M. Abed, G. Uster, et C. Kolski. Recherche d'informations personnalisées. 2000.
- [Roze et al.04] C. P. Roze, A. Anli, E. Grislin-le Strugeon, M. Abed, G. Uster, et C. Kolski. Système d'information transport personnalisé à base d'agents logiciels. *Revue Génie Logiciel*, 40 : 29–38, 2004.
- [Roze et al.05] C. P. Roze et E. Grislin-Le Strugeon. Mapis, a multi-agent system for information personalisation. *Information and Software Technology*, pp. 1–14, 2005.

- [Roze03] C. P. Roze. *Organisation multi-agents au service de la personnalisation de l'information*. Thèse de doctorat, Université de valenciennes et du Hainaut-Cambrésis, 2003.
- [Russel et al.95] Russel et Norvig. *Artificial Intelligence : A modern approach , the intelligence agent book*. Prentice Hall series in Artificial Intelligence, 1995.
- [Savy et al.01] M. Savy et C. Aubriot. *Les autorités organisatrices de transport dans les agglomérations européennes*. Rapport technique, Observatoire des politiques et des stratégies de transport en Europe : Conseil National des Transports, Septembre 2001.
- [Schulz05] F. Schulz. *Timetable Information and shortest paths*. Allemagne, Thèse de doctorat, Université Fridericiana zu Karlsruhe, Janvier 2005.
- [Shakshuki et al.03] E. Shakshuki, H. Ghenniwa, et M. kamel. An architecture for cooperative information systems. *Knowledge-Based Systems*, 16 : 17–27, 2003.
- [Shi02] G. Shi. *Data Integration using Agent based Mediator-Wrapper Architecture, tutorial report for Agent Based Software Engeneering*. Rapport technique, University of calgary, décembre 2002.
- [Steiner97] D. Steiner. Motiv-pta, personal travel assistance in germany. 1997.
- [Strahan et al.03] R. Strahan, C. Muldoon, G. O'Hare, M. Berolotto, et R. Collier. An agent-based architecture for wireless bus travel assistants. In *Proceedings The Second International Workshop on Wireless Information Systems (WIS 2003)*. Anger, France, Avril 2003.
- [Subramanian97] S. Subramanian. *Routing Algorithms For Dynamic, Intelligent Transportation Networks*. Mémoire de D.E.A., Faculty of the Virginia Polytechnic Institute and State University, 1997.
- [Sycara et al.96] K. Sycara et D. Zeng. 12 th european conference on artificial intelligence. John Wiley & Sons, Ltd, 1996.

- [Uster01] G. Uster. Développement de l'information multimodale en France : Quels leviers actionner ? *Annales des Ponts Et Chaussées : Ingénieur Science Société*, (98) : 22–26, avril-juin 2001.
- [Vidal et al.98] M. Vidal, T. Muller, P. Weinstein, et H. F. Durfee. The university of michigan digital library service market society. In K. P. Sycara et M. Wooldridge (Eds.), *the 2nd International Conference on Autonomous Agents (Agent-98)*, pp. 269–276. 1998.
- [Wang et al.03] H. Wang, J. Li, et Z. He. An effective wrapper architecture to heterogeneous data source. In *An Effective Wrapper Architecture to Heterogeneous Data Source*. 2003.
- [Wang et al.04] J. Wang et T. Kaempke. Shortest route computation in distributed systems. *Computers and operations Research*, 31 : 1621–1633, 2004.
- [Wang06] L. Wang. *Information multimodale et normalisation en France et en Chine*. Mémoire de D.E.A., Ecole Nationale des Sciences Géographiques, Mars 2006.
- [Weiss00] G. Weiss. *Multiagent Systems, A Modern Approach to distributed Artificial Intelligence*. The MIT Press, 2000.
- [Wiederhold92] G. Wiederhold. Mediators in the architecture of future information systems. *Computer*, 25(3) : 98–49, 1992.
- [Wood et al.00] M. F. Wood et S. A. DeLoach. An overview of the multiagent systems engineering methodology. In P. Ciancarini et M. Wooldridge (Eds.), *Agent Software Engineering - Proceedings of the first International Workshop on Agent-Oriented Software Engineering*. Limerick, Irlande, Springer Verlag, Juin 2000.
- [Wooldridge et al.00] M. Wooldridge, N. R. Jennings, et D. Kinny. The gaia methodology for agent-oriented analysis and design. In *Autonomous Agents and Multi-Agent Systems*, pp. 285–312. 2000.
- [Wu et al.04] Q. WU et J. Hartley. Using k-shortest paths algorithms to accommodate user preferences in the optimization of public transport travel. In *Proceedings of the 8th International Conference on Application of Advanced Technologies in Transportation Engineering*. 2004.

- [Yang et al.02] K. Yang, A. Galis, T. Mota, X. Guo, et C. Todd. Service and network management middleware for cooperative information systems through policies and mobile agents. In *EDCIS*, pp. 232–246. 2002.
- [Yin et al.05] M. Yin et m. Griss. Scateagent : Context-aware software agents for multi-modal travel. pp. 69–84. Birkhauser Verlag, 2005.
- [Zgaya et al.06] H. Zgaya et S. Hammadi. Assignment and integration of distributed transport services in agent-based architecture. In *IEEE WIC International Conference on Intelligent Agent Technology (WI/IAT'2006)*. Hong Kong chine, Décembre 2006.
- [Zidi06] K. Zidi. *Système interactif d'aide au déplacement multimodal*. France, Thèse de doctorat, Ecole Centrale de Lille, Décembre 2006.

Annexes

Etapes d'exécution du PCCDH

HAPT : Heure d'Arrivée au Plus tôt. Via AI : AI origine de l'offre

Requêtes à envoyer à: AI_A

```
SD (08:25) ----->S1
A1 (08:25) ----->A4
A1                08:28
A2                08:32
A3                08:34
A4                08:37
```

```
SD (08:25) ----->S2
A1 (08:25) ----->A6
A1                08:28
A2                08:32
A3                08:34
A4                08:37
A5                08:41
A6                08:47
```

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	true	08:37	AI_A	SD
S2	true	08:47	AI_A	SD
S3	false	99:99	-	infini
S4	false	99:99	-	infini
S5	false	99:99	-	infini
S6	false	99:99	-	infini

Nombre de Requêtes réalisées = 2

----- noeud sélectionné S1 -----

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	true	08:47	AI_A	SD
S3	false	99:99	-	infini
S4	false	99:99	-	infini
S5	false	99:99	-	infini
S6	false	99:99	-	infini

Nombre de Requêtes réalisées = 2

Requêtes à envoyer à: AI_B

```
S1 (08:37) ----->S2
B3 (08:37) ----->B2
B3                08:41
B2                08:46
```

```
S1 (08:37) ----->S3
B3 (08:37) ----->B6
B3                08:44
B4                08:47
B5                08:51
B6                08:57
```

```

S1 (08:37) ----->S4
B3 (08:37) ----->B8
B3                08:44
B4                08:47
B5                08:51
B6                08:57
B7                09:01
B8                09:06

```

```

S1 (08:37) ----->S5
B3 (08:37) ----->B9
B3                08:44
B4                08:47
B5                08:51
B6                08:57
B7                09:01
B8                09:06
B9                09:11

```

Requêtes à envoyer à: AI_A

```

S1 (08:37) ----->S2
A4 (08:37) ----->A6
A4                08:47
A5                08:51
A6                08:57

```

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	true	08:46	AI_B	S1
S3	true	08:57	AI_B	S1
S4	true	09:06	AI_B	S1
S5	true	09:11	AI_B	S1
S6	false	99:99	-	infini

Nombre de Requêtes réalisées = 7

----- noeud sélectionné S2 -----

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	true	08:57	AI_B	S1
S4	true	09:06	AI_B	S1
S5	true	09:11	AI_B	S1
S6	false	99:99	-	infini

Nombre de Requêtes réalisées = 7

Requêtes à envoyer à: AI_B

```

S2 (08:46) ----->S3
B2 (08:46) ----->B6
B2                08:53
B3                08:55
B4                08:58

```

B5 09:02
B6 09:08

S2 (08:46) ----->S4
B2 (08:46) ----->B8
B2 08:53
B3 08:55
B4 08:58
B5 09:02
B6 09:08
B7 09:12
B8 09:17

S2 (08:46) ----->S5
B2 (08:46) ----->B9
B2 08:53
B3 08:55
B4 08:58
B5 09:02
B6 09:08
B7 09:12
B8 09:17
B9 09:22

Requêtes à envoyer à: AI_A

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	true	08:57	AI_B	S1
S4	true	09:06	AI_B	S1
S5	true	09:11	AI_B	S1
S6	false	99:99	-	infini

Nombre de Requêtes réalisées = 10

----- noeud sélectionné S3 -----

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	true	09:06	AI_B	S1
S5	true	09:11	AI_B	S1
S6	false	99:99	-	infini

Nombre de Requêtes réalisées = 10

Requêtes à envoyer à: AI_B

S3 (08:57) ----->S2
B6 (08:57) ----->B2
B6 08:58
B5 09:02
B4 09:08
B3 09:12
B2 09:17

```
S3 (08:57) ----->S4
B6 (08:57) ----->B8
B6                09:08
B7                09:12
B8                09:17
```

```
S3 (08:57) ----->S5
B6 (08:57) ----->B9
B6                09:08
B7                09:12
B8                09:17
B9                09:22
```

Requêtes à envoyer à: AI_C

```
S3 (08:57) ----->S4
C7 (08:57) ----->C4
C7                09:06
C6                09:09
C5                09:13
C4                09:19
```

```
S3 (08:57) ----->S6
C7 (08:57) ----->C3
C7                09:06
C6                09:09
C5                09:13
C4                09:19
C3                09:23
```

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	true	09:06	AI_B	S1
S5	true	09:11	AI_B	S1
S6	true	09:23	AI_C	S3

Nombre de Requêtes réalisées = 15

----- noeud sélectionné S4 -----

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	false	09:06	AI_B	S1
S5	true	09:11	AI_B	S1
S6	true	09:23	AI_C	S3

Nombre de Requêtes réalisées = 15

Requêtes à envoyer à: AI_B

```
S4 (09:06) ----->S2
B8 (09:06) ----->B2
B8                09:25
```

```

B7          09:27
B6          09:30
B5          09:34
B4          09:40
B3          09:44
B2          09:49

```

```

S4 (09:06) ----->S3
B8 (09:06) ----->B6
B8          09:25
B7          09:27
B6          09:30

```

```

S4 (09:06) ----->S5
B8 (09:06) ----->B9
B8          09:17
B9          09:22

```

Requêtes à envoyer à: AI_C

```

S4 (09:06) ----->S3
C4 (09:06) ----->C7
C4          09:09
C5          09:13
C6          09:19
C7          09:23

```

```

S4 (09:06) ----->S6
C4 (09:06) ----->C3
C4          09:08
C3          09:12

```

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	false	09:06	AI_B	S1
S5	true	09:11	AI_B	S1
S6	true	09:12	AI_C	S4

Nombre de Requêtes réalisées = 20

----- noeud sélectionné S5 -----

	En cours	HAPT	via AI	Précédent
SA	false	99:99	-	infini
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	false	09:06	AI_B	S1
S5	false	09:11	AI_B	S1
S6	true	09:12	AI_C	S4

Nombre de Requêtes réalisées = 20

Requêtes à envoyer à: AI_B

```

S5 (09:11) ----->S2
B9 (09:11) ----->B2
B9          09:22

```

```

B8          09:25
B7          09:27
B6          09:30
B5          09:34
B4          09:40
B3          09:44
B2          09:49

```

```

S5 (09:11) ----->S3
B9 (09:11) ----->B6
B9          09:22
B8          09:25
B7          09:27
B6          09:30

```

```

S5 (09:11) ----->S4
B9 (09:11) ----->B8
B9          09:22
B8          09:25

```

Requêtes à envoyer à: AI_D

```

S5 (09:11) ----->SA
D4 (09:11) ----->D9
D4          09:30
D5          09:34
D6          09:40
D7          09:44
D8          09:49
D9          09:54

```

```

S5 (09:11) ----->S6
D4 (09:11) ----->D6
D4          09:30
D5          09:34
D6          09:40

```

	En cours	HAPT	via AI	Précédent
SA	true	09:54	AI_D	S5
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	false	09:06	AI_B	S1
S5	false	09:11	AI_B	S1
S6	true	09:12	AI_C	S4

Nombre de Requêtes réalisées = 25

----- noeud sélectionné S6 -----

	En cours	HAPT	via AI	Précédent
SA	true	09:54	AI_D	S5
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	false	09:06	AI_B	S1
S5	false	09:11	AI_B	S1
S6	false	09:12	AI_C	S4

Nombre de Requêtes réalisées = 25

Requêtes à envoyer à: AI_D

```
S6 (09:12) ----->SA
D6 (09:12) ----->D9
D6                09:19
D7                09:23
D8                09:28
D9                09:33

S6 (09:12) ----->S5
D6 (09:12) ----->D4
D6                09:30
D5                09:34
D4                09:40
```

Requêtes à envoyer à: AI_C

```
S6 (09:12) ----->S3
C3 (09:12) ----->C7
C3                09:27
C4                09:30
C5                09:34
C6                09:40
C7                09:44

S6 (09:12) ----->S4
C3 (09:12) ----->C4
C3                09:27
C4                09:30
```

	En cours	HAPT	via AI	Précédent
SA	true	09:33	AI_D	S6
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	false	09:06	AI_B	S1
S5	false	09:11	AI_B	S1
S6	false	09:12	AI_C	S4

Nombre de Requêtes réalisées = 29

----- noeud sélectionné SA -----

	En cours	HAPT	via AI	Précédent
SA	false	09:33	AI_D	S6
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	false	09:06	AI_B	S1
S5	false	09:11	AI_B	S1
S6	false	09:12	AI_C	S4

Nombre de Requêtes réalisées = 29

Requêtes à envoyer à: AI_D

```
SA (09:33) ----->S5
D9 (09:33) ----->D4
D9                09:40
D8                09:43
D7                09:45
```


D6 09:48
 D5 09:52
 D4 09:58

	En cours	HAPT	via AI	Précédent
SA	false	09:33	AI_D	S6
SD	false	08:25	-	SD
S1	false	08:37	AI_A	SD
S2	false	08:46	AI_B	S1
S3	false	08:57	AI_B	S1
S4	false	09:06	AI_B	S1
S5	false	09:11	AI_B	S1
S6	false	09:12	AI_C	S4

Nombre de Requêtes réalisées = 30

Le chemin multimodal est:

SD ----- S1 ----- S4 ----- S6 ----- SA

Récupération du detail:

Requête à envoyer à: AI_A

SD (08:25) ----->S1
 A1 (08:25) ----->A4
 A1 08:28
 A2 08:32
 A3 08:34
 A4 08:37

Requête à envoyer à: AI_B

S1 (08:37) ----->S4
 B3 (08:37) ----->B8
 B3 08:44
 B4 08:47
 B5 08:51
 B6 08:57
 B7 09:01
 B8 09:06

Requête à envoyer à: AI_C

S4 (09:06) ----->S6
 C4 (09:06) ----->C3
 C4 09:08
 C3 09:12

Requête à envoyer à: AI_D

S6 (09:12) ----->SA
 D6 (09:12) ----->D9
 D6 09:19
 D7 09:23
 D8 09:28
 D9 09:33

Résumé

Afin d'éviter au voyageur de consulter plusieurs sites web d'opérateurs de transport en commun pour planifier son déplacement, ce travail vise à concevoir un Système d'Information Coopératif de Mobilité (SICM) pour l'aide au déplacement multimodal. Il s'agit d'automatiser cette démarche de recherche et de composition d'itinéraires, pour fournir une information multimodale via un système intégrateur, en s'appuyant sur la théorie des systèmes multi agents (SMA) pour l'intégration et la médiation des systèmes d'information des différents opérateurs de transport.

Pour produire l'information multimodale et multi-opérateurs nécessaire à l'aide au déplacement, le SICM doit accéder aux différents systèmes d'information des opérateurs de transport et intégrer des résultats de recherche qui sont générés par les différents algorithmes des différents opérateurs. Dans cette approche, le SICM est un intergiciel (middleware) qui devient un client parmi d'autres usagers des systèmes d'information existants.

Le SICM devient alors l'intermédiaire entre les différentes sources d'informations hétérogènes et distribuées d'une part et les clients d'autre part. Ce système doit être capable à la fois de trouver la bonne source d'information pour l'interroger selon les différentes requêtes des utilisateurs, et de regrouper les informations de manière cohérente pour répondre aux requêtes. Pour fournir un itinéraire composé mais surtout optimisé selon les critères de l'utilisateur, le recours à des algorithmes de plus courts chemins distribués « en-ligne », et adaptés à des graphes dynamiques (dépendant du temps) a été retenu afin de réaliser ce moteur de recherche et de composition d'itinéraires multimodaux en ligne.

Mots-clés

Information voyageur - Transport multimodal - Information multimodale - Système d'Information - Systèmes multi-agents - Recherche d'itinéraires

Abstract

To plan his travel, a traveller has to consult several web sites of different public transport operators. To avoid this time consuming task, this work consists in conceiving a Mobility Cooperative information system (SICM) providing a multi-modal and a multi-operators travel information. This integration system automates the itineraries search and the multi-operators routes composition. Its design is based on the multiagent system theory (MAS). The SICM tries, in fact, to make the existing operators' information systems cooperating efficiently together, so that it can provide users with the optimized route to follow, by compiling the needed information from the different operators information sources.

In this approach, the SICM is a middleware which becomes a customer among other users of the existing information systems. It can be considered as a mediator between the various distributed information sources on the one hand and the travellers on the other hand. The system should be able, at the same time, to find the needed information sources which are able to answer an itinerary request, and to gather this information in a coherent way to compose an optimized itinerary. To provide an optimized route, according to the criterion of the user, distributed and time-dependent shortest path algorithms were adopted and adapted to realize an on-line itinerary composition.

Key-words

Travel-information - Multimodal Transport - Multimodal information - Information System - Multi-Agent Systems - Itineraries search