# Serveur Agent d'Information : application à l'information des voyageurs

**Flavien Balbo*,** — Mahdi Zargayouna*,****

* *Inrets - Gretia, Institut National de Recherche sur les Transports et leur Sécurité.*
*2, avenue du Général Malleret-Joinville,*
*F-94114 Arcueil cedex.*

** *Lamsade, Paris Dauphine University.*
*Place Maréchal de Lattre de Tassigny,*
*75775 Paris Cedex 16, France.*

*balbo@lamsade.dauphine.fr ; zargayou@inrets.fr*

RÉSUMÉ. *Dans cet article, nous proposons une architecture de serveur informationnel fondée sur le paradigme multi-agent. Notre proposition s'appuie sur le modèle EASI (Environnement comme support Actif des Interactions) qui met en œuvre dans le domaine multi-agent le concept d'écoute flottante. Il s'agit de permettre aux agents d'évoluer au sein d'un environnement fortement interactionnel. Nous avons appliqué notre proposition au domaine du transport et plus particulièrement de l'information des voyageurs. Nous montrons dans cet article, que notre proposition répond à la problématique du domaine en permettant à chaque voyageur d'organiser sa recherche ainsi que la gestion de la mise à jour des informations pertinentes pour son voyage.*

ABSTRACT. *In this paper, we propose architecture of an informational Server founded on the multi agent paradigm. Our proposition is based on the EASI model (Environment as an Active Support for Interactions) which implements the concept of the mutual awareness in the multi agent domain. The goal is to permit agents to evolve in a strongly interactional environment. We have applied our proposition to the transport domain and more particularly in the traveller information. We demonstrate in this paper that our proposition answers to the domain problems by permitting to every traveller to organise his research and to manage her relevant trip information update.*

MOTS-CLÉS : *Plateformes informationnelles, transport, Médiation de sources d'informations*
KEYWORDS: *informational platforms, transport, mediation of information sources.*

## 1. Introduction

Avec le développement des réseaux de communications, nous observons ces dernières années une multiplication des sources et des vecteurs d'information. Face à cette multiplication des possibilités, l'utilisateur a besoin de se faire aider, c'est pourquoi de nouvelles problématiques sont apparues comme la délégation de traitement dans la recherche d'informations. Le paradigme multi-agent a trouvé une place naturelle dans ce domaine comme le montre par exemple le nombre important de recherches dans le domaine du commerce électronique où la recherche d'information et son traitement sont réalisés par un agent prenant en compte les besoins d'un utilisateur (Bichler, 2000). Pour trouver les informations pertinentes pour l'utilisateur, chaque agent doit interagir avec les différentes sources qui peuvent être elles mêmes des agents. Dans le domaine multi-agent, la gestion des interactions est souvent assimilable à une gestion des compétences. La problématique est de proposer le meilleur compromis entre répartition des compétences et complexité de recomposition (qui fait quoi ? comment trouver le meilleur partenaire ? …). Du point de vue de la gestion des informations, la conception de systèmes multi-agents a souvent suivi cette voie : les serveurs d'informations sont assimilés à des agents ayant des compétences dans un domaine (ou plusieurs) et l'interaction avec ces agents s'appuie sur des protocoles classiques dans le domaine comme les agents médiateurs (Decker *et al.*, 1997) ou les réseaux d'accointances (Knoblock *et al.*, 1994). De même, les langages de haut niveau soutiennent ce principe en intégrant des performatifs tels que *Subscribe* (Finin *et al.*, 1994) (FIPA ACL, 1999) ou *Broadcast* (Finin *et al.*, 1994) qui organisent la relation entre les agents producteurs et consommateurs d'information.

Le parallèle compétence / serveur atteint ses limites lorsque l'intérêt porté à une information ne dépend pas de sa nature mais de sa valeur. Par exemple, la conception d'une application devant prendre en compte un contexte informationnel fortement dynamique devra proposer un mécanisme de traitement de l'information en fonction de son contenu. Si nous prenons l'exemple de la gestion des informations dans un poste de régulation aérienne, il a été montré (Salembier, 1994) qu'une partie des interactions relevait du concept de *l'écoute flottante* ou *mutual awareness* (Dugdale *et al.*, 2000). Les opérateurs ont l'ensemble des informations du poste de régulation à leur disposition et ils dirigent leur attention en fonction de l'intérêt d'une information par rapport à leur activité. L'assimilation compétence / serveur ne répond pas à cette problématique puisque les sources d'information sont connues et c'est l'information en tant qu'unité de traitement qui doit être gérée. Le principe de la mise en commun des informations puis d'une sélection locale par les agents en fonction de leur activité a également été testé dans le cadre de la résolution de problèmes (De Loor *et al.*, 2003). L'application choisie concerne la gestion distribuée d'emploi du temps et la mise en commun des informations est réalisée par diffusion. Les différents tests réalisés montrent que si les objectifs de résolution du problème sont atteints, le recours aux diffusions pénalise fortement l'efficacité de la proposition.

C’est dans cette problématique de gestion des informations comme unité primaire de traitement que s’inscrit notre proposition. Afin de permettre une personnalisation de l’accès aux informations en fonction du contexte des utilisateurs, nous proposons une architecture de serveur d’informations permettant un paramétrage dynamique par ses utilisateurs. Afin d’illustrer notre proposition, nous avons choisi le domaine du transport et plus particulièrement le domaine de la gestion des informations voyageurs. Notre proposition consiste à définir un système d’information des voyageurs reposant sur le « moment de pertinence des informations ». L’évaluation de ce critère est le résultat d’un compromis entre les besoins et préférences du voyageur mais également des besoins de l’émetteur de l’information.

Afin de soutenir notre propos, nous avons suivi le plan suivant : dans une première partie, nous présentons le domaine de l’information voyageur. Dans une seconde partie, nous présentons notre proposition d’architecture d’un Serveur Agent d’Information Voyageurs (SAIV). Dans une troisième partie, nous donnerons nos premiers résultats ainsi qu’un exemple de mise en œuvre de notre proposition. Enfin, la dernière partie nous permettra de conclure et de présenter les perspectives de nos travaux.

## 2. Le domaine de l’information voyageur

Dans le domaine du transport, nous pouvons anticiper une augmentation des exigences du voyageur en matière de qualité de l’information. En effet, les sources d’information sont de plus en plus nombreuses et l’évolution des technologies permettent théoriquement à chacun de recevoir de l’information quelque soit le lieu où il se trouve. La conception d’un système d’information voyageur (SIV) doit répondre à ces exigences (Predit, 2001) en aidant le voyageur dans son choix modal et en lui facilitant l’usage des réseaux. Par conséquent, un SIV doit proposer de l’information pré voyage en donnant l’offre globale de tous les modes de transport sur un déplacement donné. Le SIV doit également donner de l’information pendant le voyage en accompagnant le voyageur et en l’informant à bon escient. La conception d’un SIV remplissant ces objectifs aura pour avantage d’accroître la confiance de l’usager dans son usage des réseaux et constitue un facteur de fidélisation.

Constatant l’importance de l’information du voyageur dans leur relation avec leur clientèle, les réseaux de transport ont proposé une réponse adaptée à leur besoin comme nous le décrirons au paragraphe 2.1. Cependant, cette réponse est hétérogène et disséminée et les systèmes multi-agents sont utilisés pour proposer une réponse agrégée comme nous le décrirons au paragraphe 2.2.

### *2.1 Approche exploitant*

Le rapport (Predit, 2001) a mis en évidence l’information multimodale comme un des quatre composants à prendre en compte afin de simplifier l’usage des transports. L’enjeu pour un SIV est de proposer le mode optimal de transport pour le déplacement

souhaité et d'aider à sa réalisation. Ainsi, les réseaux de transport ont constitué une offre d'information dite passive ou interactive. Une information est dite passive, si elle est donnée sans possibilité de compléments, c'est par exemple le cas d'un panneau à messages variables, d'information sur papier ou de tableau d'affichage. A contrario, une information est dite interactive si elle est obtenue à l'initiative du voyageur et contient des informations personnalisées, c'est par exemple le cas d'un serveur Web ou de l'utilisation d'un assistant personnel. Cependant, de part leur source, l'information fournie est généralement monomodale (concerne un seul mode de transport). Chaque opérateur présente son offre en information pour répondre aux besoins de sa clientèle et se limite à son territoire de compétence.

La gestion par un opérateur de son propre système d'information a pour avantage de faciliter la mise à jour des informations assurant ainsi aux utilisateurs un service de qualité. De plus, l'information des voyageurs constitue souvent une utilisation d'informations existantes utilisées par d'autres composants du système d'information de l'opérateur. Par exemple, dans le cas des transports en commun, l'information fournie aux voyageurs correspond à des données utilisées par le Système d'Aide à l'Exploitation (SAE). Ces données ne sont pas conçues pour être intégrées au sein d'un système d'information concevant une offre globale pour le voyageur. Par conséquent, l'utilisation du système d'information suppose une adaptation de l'utilisateur aux choix de l'opérateur et, si le besoin est multi opérateurs, alors il y aura autant d'adaptations que de sources d'information. Cette hétérogénéité des sources rend difficile une gestion conjointe des informations et suppose une intervention importante de l'utilisateur. C'est pourquoi la conception d'un voyage demeure une tâche complexe (Brien *et al.*, 1998). Si cette tâche s'avère fastidieuse pour un être humain, l'hétérogénéité des sources d'information accroît la complexité pour un agent artificiel. Pourtant, certains traitements pourraient être automatisés comme l'optimisation des combinaisons horaires entre les différentes parties d'un trajet ou l'évaluation des offres tarifaires. C'est dans cet objectif d'automatisation et de conception de services avancés que le paradigme multi-agent offre des solutions.

### *2.2 Approche multi-agent*

Le domaine du transport est propice à l'utilisation des modèles développés dans le domaine multi-agent. Les propriétés d'autonomie, d'aptitudes sociales et de réactivité qui sont usuellement associées aux agents intelligents (Ferber, 1995) sont fréquemment mises à contribution pour développer des systèmes adaptés au domaine du transport (Schleiffer, 2002). Dans le domaine de l'information voyageur, il est possible de définir un ensemble de fonctions qu'un agent peut accomplir pour aider un utilisateur (Petit-Rozé et al., 2002). Sa première fonction est de collecter l'information sur des systèmes hétérogènes et distribués. En effet, l'utilisateur doit pouvoir formuler son besoin sans connaître les sources d'information pouvant y répondre. Afin de constituer une réponse personnalisée, la seconde fonction consiste à intégrer les données obtenues. Le voyageur doit pouvoir spécifier ses préférences et obtenir une (des) réponse(s) en

conséquence. Enfin, la dernière fonction consiste à assurer le suivi des informations afin de surveiller le bon déroulement du déplacement. Le voyageur doit pouvoir être averti de tout évènement susceptible d'interférer avec son déplacement.

C'est cette adéquation entre la problématique du domaine de l'information voyageur et les caractéristiques admises des agents qui est à l'origine des travaux de la FIPA (Foundation for Intelligent Physical Agents) dont un des objectifs est de montrer la faisabilité des technologies agents (O'Sullivan *et al.*, 2000).

Le système permet à un utilisateur humain de réserver un voyage simplement en indiquant le détail de son besoin à son PTA (Personal Travel Agent), qui prendra les conditions et les enverra à un TBA (Travel Broker Agent). Cet agent décompose le voyage en série de segments et questionne les TSA (Travel Service Agent) qui représentent les fournisseurs de service (lignes aériennes, chaînes d'hôtel …). Les réponses des TSA constituent des morceaux de réponses que le TBA doit agréger. Il envoie alors un voyage empaqueté au PTA pour l'évaluation. Cet agent regardera les préférences et contrôlera la validité de la proposition. Si l'offre n'est pas susceptible d'être reçue, alors elle est rejetée, autrement elle est soumise à l'utilisateur. L'utilisateur valide l'offre et le TBA procède à la réservation des différents segments du voyage.

L'organisation proposée par la FIPA est efficace pour l'obtention d'informations pré voyage. Elle propose une solution au problème de collecte et personnalisation de l'information et une fonction avancée en automatisant la phase de négociation. Si ces traitements sont indispensables à l'automatisation de l'activité d'une agence de voyage virtuelle, ils sont inutiles dans le domaine de l'information voyageur au quotidien. Ainsi, pour un voyageur sur le réseau multimodal d'une agglomération, le problème n'est pas d'identifier les sources d'information mais de gérer dynamiquement son déplacement. L'activité des agents ne consiste pas à gérer des « compétences » (par exemple les agents TSA dans la proposition FIPA) mais à gérer des informations en tant qu'unité de traitement. Le problème consiste à définir pour chaque voyageur le moment de pertinence d'une information. Par conséquent, nous proposons la conception d'un lieu d'échange des informations permettant à chaque agent de construire dynamiquement son contexte interactionnel i.e le contexte où il souhaite entrer en interaction avec les autres agents de l'environnement et sous quelles conditions.

## 3 Modélisation d'un Serveur Agent d'Information

Le serveur agent d'information que nous proposons s'appuie sur le modèle EASI (Environnement comme Support Actif de l'Interaction). EASI constitue une généralisation du modèle ESAC (Balbo, 2000) (Environnement comme Support Actif de Communication) qui a été utilisé dans le cadre d'une application concernant la régulation d'un réseau de bus (Système SATIR, (Balbo, 2001)).

Dans une première partie, nous présentons les bases du modèle EASI puis dans une seconde partie, nous présentons son utilisation pour la conception d'un serveur agent d'information.

### *3.1 Fondement : modèle EASI*

Le principe d'un milieu commun aux agents est central dans la communauté des agents réactifs. Dans ce paradigme, les agents possèdent des capteurs afin de percevoir l'environnement et des effecteurs afin d'agir sur ce dernier. Etendue aux agents cognitifs, le milieu commun est un espace d'interaction que chaque agent peut modifier par ses propres interventions (i.e. émission de message) et percevoir au moyen de capteurs. Le modèle EASI permet à des agents cognitifs d'utiliser l'environnement pour échanger des messages. Un SMA n'est donc plus conçu comme une somme de communications organisées par des protocoles, mais comme un espace de communication où chaque message est perceptible indépendamment des besoins initiaux de l'émetteur et du récepteur du message ainsi que de leur protocole en cours.

Les messages échangés doivent pouvoir être complexes (agents cognitifs) et les agents doivent percevoir dans l'environnement les messages dont ils sont récepteurs (agents réactifs). Les messages échangés peuvent être adressés ou pas. L'utilisation de l'environnement conditionne la réception d'un message aux capacités de perception des agents.

La communauté des agents réactifs a produit une modélisation des agents, de leur perception de l'environnement à leur réaction. En fonction de leurs capacités, ce cycle est plus ou moins long. Nous proposons d'étendre ces travaux en modélisant les filtres de perception de l'environnement par des agents cognitifs.

Pour la communauté des agents réactifs (un agent tropique (Ferber, 1995b)), un agent se définit classiquement comme un triplet :

$$a = < P_a, Percept_a, Reflexe_a >$$

Avec :

- $P_a$ : ensemble des percepts associés à un agent,
- $Percept_a$ : fonction de perception qui associe un percept à chaque état du monde,
- $Reflexe_a$ : fonction qui associe une opération à un percept.

Par extension, un agent cognitif percevant les communications transitant par l'environnement se définit comme :

$P_a$ : messages déposés dans l'environnement et accessibles à l'agent *a*. Théoriquement l'ensemble des messages déposés dans l'environnement est accessible aux agents présents.

$Percept_a$ : ensemble des filtres de perception des communications permettant à l'agent *a* de recevoir des messages. La définition des filtres permettant la perception des communications définit le sous-ensemble des messages perceptibles par un agent.

$Reflexe_a$ : activation du processus de raisonnement en fonction du message reçu (contenu et filtre). La prise en compte du message par l'agent sera dépendante du contenu du message mais également du filtre ayant permis sa réception.

Nous proposons de modéliser l'environnement par des filtres permettant d'adresser des messages à des agents. Nous noterons $\Omega$ les éléments présents dans l'environnement :

$$\Omega = <\Omega_A, \Omega_M>$$

Avec :

- $\Omega_A$ : ensemble des agents du SMA.
- $\Omega_M$ : ensemble des messages du SMA.

Chaque élément présent dans l'environnement est identifiable par un ensemble de propriétés accessibles par l'environnement. Nous donnons ici les définitions des composants du modèle EASI : Agent, Message et Filtre.

**Définition 1 : un agent**

Soit *A* une catégorie d'agent et *a* un agent de *A* alors

$$\forall\ a \in A,\ a = \{\ p(a) \mid p \in \mathsf{P}(A, D_\Omega)\}$$

On notera $\mathsf{P}(\Omega_A, D_\Omega)$ l'ensemble des propriétés des agents du SMA. Chaque propriété p est une fonction qui, pour une entité (agent ou message), associe une valeur sur le domaine de définition $D_p$ si la propriété est définie pour l'entité concernée et *null* sinon. $D_\Omega$ est constitué par l'union des domaines de définition des propriétés des entités de $\Omega$. $\mathsf{P}(A, D_\Omega)$ est le sous ensemble de propriétés concernant les agents de la catégorie *A*.

En résumé, une catégorie d'agent définit un ensemble d'agents par un sous-ensemble de propriétés communes. Un agent est définit comme membre d'une catégorie et se différencie par les valeurs de ses propriétés.

**Définition 2 : un message**

$$\forall\ m_o \in \Omega_M,\ m_o = <e_o, r_o, sujet_o, \bigcap_{l=1}^{n} C_{ijk}^{l}, t_o>$$

Avec :

- $e_o$ : Identifiant de l'émetteur du message,
- $r_o$ : Identifiant du récepteur,
- $sujet_o$ : Sujet du message,

- $C_{ijk}$ = <$p_i$, $f_j$, $v_k$>
  - $p_i \in P(\Omega_A, D_\Omega)$
  - $f_j : D_{pi} \times D_{pi} \rightarrow \{Vrai, Faux\}$
  - $v_k \in D_{pi}$
- $D_{pi}$ : Domaine de définition de la propriété $p_i$,
- $\bigcap_{l=1}^{n} C_{ijk}^{l}$ : Conditions à remplir pour recevoir le message,
- $t_o$ : Durée de vie maximale du message.

Une condition $C_{ijk}(\omega)$ est vraie si :

$$\omega \in \Omega \wedge f_j(p_i(\omega), v_k)$$

Un message est envoyé par un émetteur $e_o$ à un récepteur $r_o$ (connu ou pas), ses besoins pour la recherche du récepteur sont contenus dans les conditions. La durée de vie du message (date au-delà de laquelle le message est périmé) dépend de la variable $t_o$.

Cette définition constitue la sémantique des messages qui est commune à l'ensemble des agents. La description d'une condition devient par construction une propriété d'un message (voir exemple au paragraphe 4.2).

**Définition 3 : un filtre**

On notera $F_n$ le n$^{ième}$ filtre de l'environnement.

$$m \in \Omega_M, a \in \Omega_A, \quad F_n(m, a) = \left(\bigcap_{l=1}^{l_a} C_{ijk}^{l}(a)\right) \bigcap \left(\bigcap_{l=1}^{l_m} C_{ijk}^{l}(m)\right) \bigcap \left(\bigcap_{l=1}^{l_e} C_{ijk}^{l}\right)$$

Avec :

- $\left(\bigcap_{l=1}^{l_a} C_{ijk}^{l}(a)\right)$ : Conditions portant sur l'agent récepteur du message,
- $\left(\bigcap_{l=1}^{l_m} C_{ijk}^{l}(m)\right)$ : Conditions portant sur le message traité,
- $\left(\bigcap_{l=1}^{l_e} C_{ijk}^{l}(e_l)\right)$ : Conditions portant sur les autres entités identifiant l'interaction,
- $l_a$, $l_m$, $l_e$ : nombre de conditions pour chaque catégorie d'entité.

Un filtre met en relation un agent *a* avec un message *m*. Un agent peut concevoir un filtre avec des conditions portant à la fois sur les agents et sur les messages. Cette possibilité permet toutes les perspectives dans la gestion de ses interactions. Par exemple, un agent A peut construire un filtre lui permettant de recevoir les messages de l'agent B en conditionnant son usage à l'événement interactionnel suivant : l'agent C a envoyé un message à l'agent D. Par conséquent, il n'est pas possible de représenter les paramètres d'un filtre autrement que par des ensembles de conditions.

Appliquée au domaine de l'information des voyageurs, le modèle EASI permet à chaque voyageur de constituer son propre domaine d'intérêt parmi l'ensemble des informations disponibles. L'intérêt du voyageur peut porter sur d'autres informations que celles relatives à la qualité de son déplacement. Notre modèle permet à chaque voyageur de concevoir les filtres lui permettant de représenter son intérêt pour les différentes sources d'information. Sur la Figure 1, l'attention d'un voyageur est représentée par une zone recouvrant l'ensemble des informations auxquelles le voyageur souhaite accéder. Le filtre permet au voyageur de préciser la nature des informations qui l'intéressent ainsi que le contexte dans lequel il souhaite la recevoir (sa position par exemple). Ainsi, ce voyageur sera informé de l'évènement 1 et non de l'évènement 2.

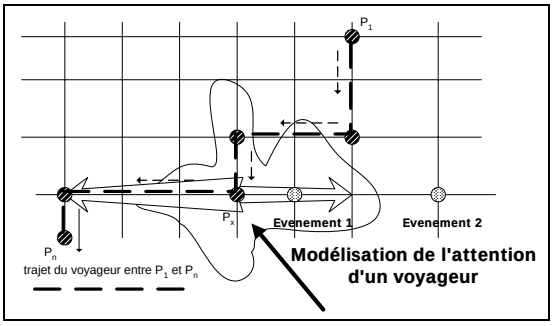


**Figure 1 :** *représentation de l'attention du voyageur*

### *3.2 Utilisation du modèle EASI dans le cadre d'un serveur d'information*

Le modèle interactionnel proposé par EASI, permet de concevoir un serveur d'information paramétré par ses utilisateurs. Si les systèmes d'information sont adaptés à la satisfaction de besoin ponctuel (modèle requête/réponse), la gestion du suivi d'information suppose la mise en œuvre de traitements supplémentaires. Ces traitements seront d'autant plus complexes que les sources d'information sont distribuées et que la gestion du suivi suppose une comparaison des informations disponibles. Le modèle EASI permet une mise en commun des informations disponibles et mise à jour par les serveurs. Il ne s'agit pas de dupliquer les données proposées par des serveurs d'information mais d'organiser leur utilisation dans un contexte défini.

Si nous étendons le principe de l'utilisation de l'environnement pour la conception d'un serveur d'informations, il est nécessaire de préciser l'utilisation de chacun des composants : 1) les agents ; 2) les messages ; 3) les filtres.

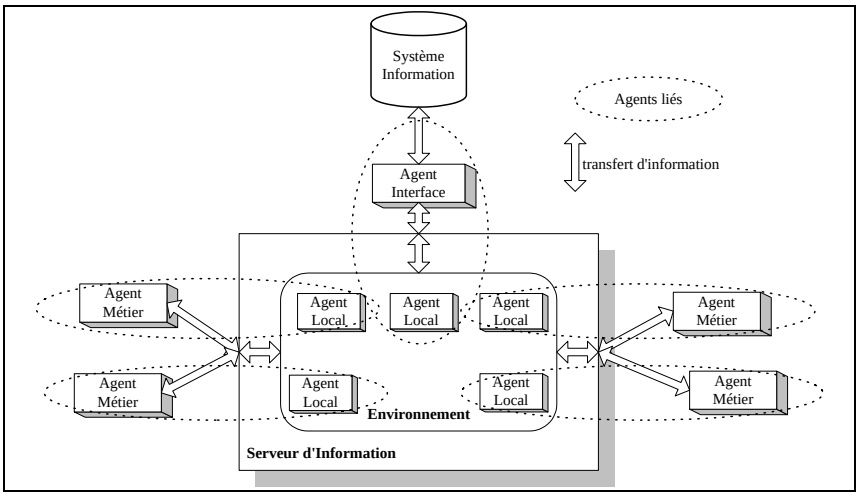


**Figure 2 :** *Architecture du Serveur Agent d'Information*

Nous avons identifié trois catégories d'agents. La première concerne les agents *Interface*, son rôle est de faire le lien entre un système d'information existant et le serveur d'information. Cet agent devra collecter l'information nécessaire à l'application et/ou répondre aux demandes issues du serveur. Les informations envoyées par cet agent seront déposées dans l'environnement et respecteront la syntaxe des messages présentée au paragraphe précédent. La seconde catégorie concerne les agents *Métier* qui sont spécifiques à l'application. Ces agents peuvent être récepteurs et/ou émetteurs d'informations. Contrairement aux agents *Interface* dont le rôle est d'intégrer au sein de l'application des sources de données existantes, ces agents ont été conçus afin d'exploiter le serveur d'information. A ces deux catégories d'agent qui sont physiquement distants s'ajoute la catégorie agent *Local*. Cet agent est le représentant d'un agent *Interface* ou *Métier* au sein de l'environnement. Son rôle est de gérer les filtres pour l'agent distant (*Interface* ou *Métier*) auquel il est associé. L'architecture du serveur d'information doit permettre de distribuer l'information entre la structure émettrices / réceptrices distantes (Agents *Interface* et *Métier*) et la structure organisatrice (Agents *Local*)

Appliqué au domaine de la gestion de l'information du voyageur, le SAIV doit permettre de faire le lien entre les sources d'information et les utilisateurs d'un réseau multimodal donné. Un réseau multimodal est un réseau composé de différents modes de transport (bus, métro, voiture personnelle etc.). Nous avons défini deux catégories d'agents *Métier*, la première concerne les agents liés à l'utilisateur (que nous appelons *MPTA* pour Mini Personal Travel Agent) tandis que la seconde concerne les agents liés aux services que notre serveur d'information permet d'envisager (que nous appelons agent *Service*).

L'utilisateur étant mobile, nous avons procédé à une répartition des compétences afin de limiter les interactions avec le SAIV. Par conséquent, L'agent *Local* (que nous appelons *PTA* pour Personal Travel Agent) représentant d'un agent *MPTA* sur le serveur devra gérer les besoins en information (gestion des filtres) mais également procéder à un traitement local de ces informations. Par exemple, si une perturbation est détectée sur le trajet d'un voyageur, celui-ci ne doit être informé que si son déplacement est fortement perturbé.

Un agent *Service* doit être utilisateur du serveur d'information et non seulement un fournisseur d'information. Ainsi, un agent lié à un service permettant de calculer un itinéraire appartient à la catégorie agent *Interface* s'il est capable de répondre à des requêtes de demande d'itinéraire. Cependant, si ce même service peut recevoir les informations concernant les perturbations et en tenir compte dans le calcul d'un itinéraire alors il appartient à la catégorie agent *Service*.

## 4 Premiers Résultats

Dans cette section, nous présentons l'architecture de l'application ainsi qu'un exemple d'utilisation illustrant l'échange de flux d'informations entre les différents acteurs et le serveur d'information de voyageurs.

Notre choix de déployer notre application sur le Web nous est dicté par la nécessité d'offrir un service accessible aux utilisateurs mobiles, mais aussi par notre souci d'interagir avec des applications informatiques, systèmes d'information distants. Ce deuxième volet sera basé essentiellement sur des techniques issues du domaine des services Web.

Le deuxième choix concerne l'interaction avec les *MPTA.* La seule exigence sur ces supports mobiles est leur support de l'échange de données html. Autrement dit, aucune capacité de calcul ne leur est exigée.

Dans ce qui suit, nous passons à la description de l'architecture logicielle de notre application Web du serveur agent d'information de voyageurs.

### *4.1 Description de l'architecture logicielle*

Lors de la description de l'utilisation du modèle EASI dans le cadre d'un serveur d'information (section 3.2), nous avons identifié trois types d'agents en interaction avec le système, à savoir les agents *Interface*, les agents *Métier* et les agents *Local*. Au sein des agents *Métier*, nous avons distingué deux catégories : la première concerne les agents liés à l'utilisateur (*MPTA*). La seconde concerne les agents liés aux services que notre serveur d'information permet d'envisager (agent *Service*).

Bien que d'un point de vue théorique, l'interaction avec les agents *Métier* se base sur le même principe, techniquement, selon la catégorie d'agent, cette interaction suit un schéma différent. D'un côté, l'interaction des agents *MPTA* avec le serveur sera une

séquence de requêtes/réponse http. De l'autre, l'interaction avec les agents *Service* sera basée sur l'échange de messages suivant un protocole de haut niveau au dessus de la couche http.

Notre architecture est par conséquent logiquement divisée en une face client (orientée *MPTA*) et une face service (orientée agent *Service*). Dans chacun de ces volets, un problème est à résoudre : l'échange asynchrone de données. En effet, un utilisateur devra recevoir des informations relatives à son trajet (trafic par exemple) sans qu'il en fasse la demande, ces données venant d'une manière ad hoc et en temps réel. Aussi, un service doit pouvoir recevoir des messages (demande de plan par exemple) et envoyer des messages au serveur (perturbation par exemple) d'une manière asynchrone et non en réponse à une requête donnée. Seulement, le protocole http étant un protocole basé sur un paradigme requête/réponse, il est typiquement synchrone.

Du côté client, l'échange de messages asynchrones est assuré par la création d'intermédiaire : à la connexion d'un *MPTA*, une page xml est créée sur laquelle il est redirigé. Dorénavant, les messages qui lui sont destinés seront redirigés vers cette page.

Du coté service, le principe est différent. L'échange asynchrone de messages est assuré via une api issue de la communauté des services Web, JAXM (Java Api for Xml Messaging) (JAXM, 2002), avec utilisation d'un 'provider'. Le principe général est le même, il s'agit du passage par un intermédiaire (le 'provider') afin de contourner l'exigence de synchronisation du protocole http. L'émetteur crée un message SOAP (Simple Object Access Protocol) (SOAP 2001) dans lequel sont spécifiées –entre autre- sa propre adresse ainsi que celle du récepteur et l'envoie à son 'provider' local. Il n'attend par conséquent pas la réponse de son interlocuteur avant d'engager d'autres traitements. Son 'provider' localise alors le récepteur et lui envoie le message qui sera reçu par le récepteur (classe JAXMServlet, méthode onMessage). La figure 3 montre l'architecture de l'application Web.

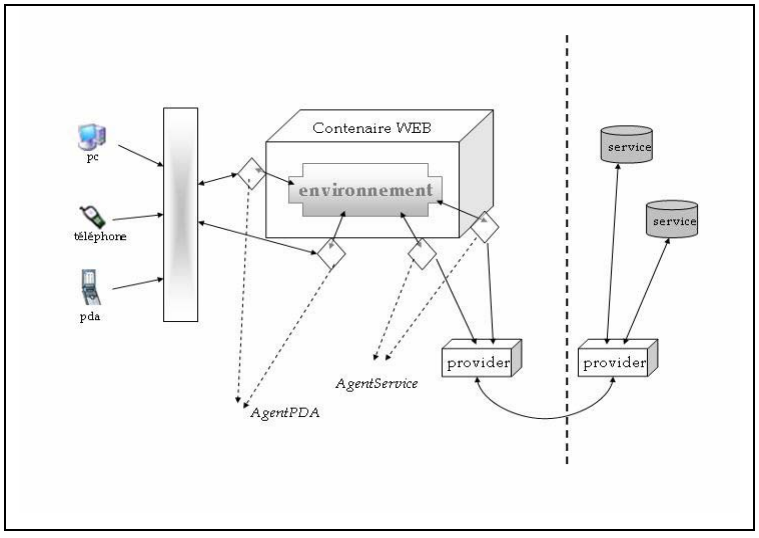

**Figure 3 :** *Architecture logicielle Système d'Information Agent*

Dans ce qui suit, nous présentons l'acheminement d'un message, depuis la connexion du *MPTA* jusqu'à l'arrivée à destination du message émis par l'agent *PTA* le représentant, ainsi que le chemin inverse d'un message issu du service jusqu'au *MPTA*. Pour ce faire, nous nous fondons sur un exemple représentant une utilisation possible du système. Par cet exemple, notre objectif est de montrer un exemple d'acheminement de messages et non toutes les possibilités d'interaction au sein de notre système.

### *4.2 Exemple d'utilisation*

Le contexte de notre exemple est le suivant : au sein de notre système, figurent un agent représentant d'un service de planification multimodale de trajet et un agent représentant d'un service de trafic responsable de l'émission de messages relatifs aux incidents, bouchons etc… Ces agents sont persistants, puisqu'ils sont constamment en relation avec le système informatique distant fournisseur de service. En revanche, les agents *PTA* représentants des *MPTA* dans notre système sont volatiles, créés à la connexion d'un utilisateur et écrasés à la fin de sa session i.e. lors de son arrivée à destination.

Nous avons développé le service de planification qui est une application Web distante supportant JAXM et dont le rôle est d'abord de recevoir la demande de plan (sous forme d'un message SOAP) puis de se connecter sur un site Web offrant un service de planification. Le résultat est enveloppé dans un message SOAP et renvoyé à notre service de planification. L'intérêt du passage par une application Web 'traditionnelle' i.e offrant un service en ligne, est de démontrer la transparence de cette connexion (synchrone) vis à vis du serveur d'information de voyageurs et de l'utilisateur final. En effet, l'exigence de synchronisation avec le site Web (protocole http) est levée en passant par notre service de planification supportant JAXM.

La figure 4 illustre notre exemple. Chronologiquement, le *MPTA* se connecte au serveur d'information de voyageurs, son agent Local représentant est créé. Ensuite, l'utilisateur est prié de spécifier son lieu de départ ainsi que son lieu d'arrivée. Une fois ces informations entrées, sa page xml dédiée est créée et il y est redirigé dans l'attente de réception de messages. Sa demande de plan est alors enveloppée dans un message et déposée dans l'environnement à destination de l'agent *Local* représentant le service de planification, le message aura dans ce cas la structure suivante :
<id-agent_représentant, id-agent_Local, « demande de plan », arrêt_départ, arrêt_arrivée, ∞>. Arrivé à la hauteur de l'agent, le message est intercepté. Le rôle de cet agent est l'interfaçage entre le serveur et l'application distante. En effet, recevant une demande de plan selon un format spécifique, il a pour rôle de transformer la

demande et de l'envelopper dans un message SOAP avant de l'envoyer via JAXM au service de planification distant.

Comme nous l'avons avancé en introduction, le service de planification distant se connectera via une requête http à un site Web à la demande de plan et en reçoit la réponse. La procédure inverse est alors déclenchée jusqu'à l'arrivée du message au niveau de l'agent *PTA* qui renverra la réponse (fichier xml) vers la page Web dédiée à l'utilisateur.

Outre la localisation de la page dédiée à l'utilisateur et la redirection du plan, l'agent *PTA* analyse le document xml reçu qui est composé de tronçons d'itinéraires ; un tronçon d'itinéraire étant un itinéraire continu (monomodal) sans correspondances. A partir de ce document, l'agent *PTA* génère un filtre par tronçon d'itinéraire destiné à recevoir les informations relatives aux incidents qui surviendraient sur son parcours, il y aura donc autant de filtres que de tronçons. Ces filtres sont déposés dans l'environnement. Ainsi, les conditions de réception d'un message sont paramétrées par le contexte de l'utilisateur. Ce dernier recevra un message d'alerte relatif à son parcours si et seulement si cette alerte concerne le plan qu'il a reçu. Chaque filtre déposé dans l'environnement aura donc la structure suivante :
<id-émetteur, ligne_concernée, arrêt_départ, arrêt_arrivée, ∞>.

L'acheminement d'un message du *MPTA* vers le service et inversement suit le même itinéraire décrit plus haut.

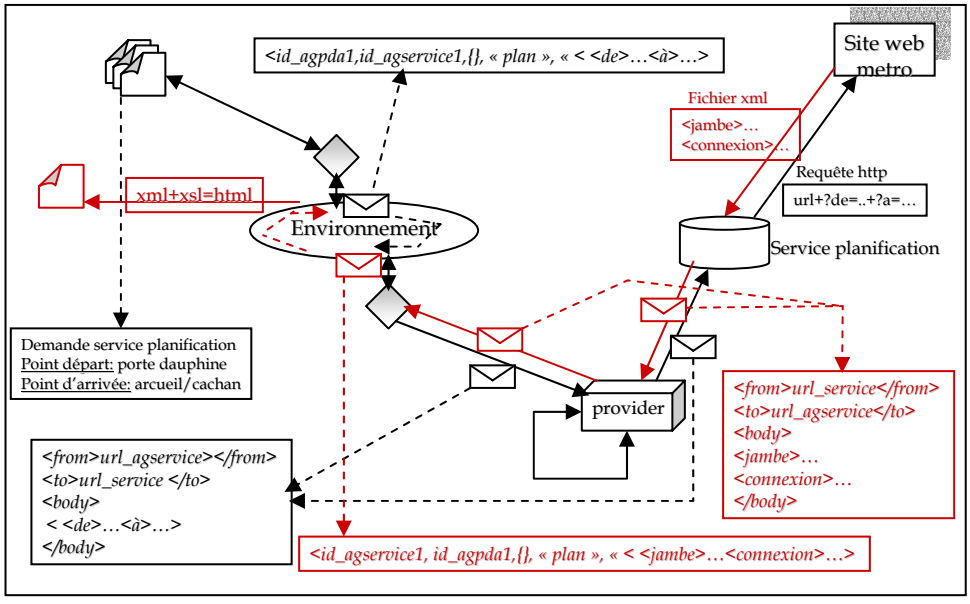


**Figure 4 :** *Acheminement d'un message*

## 5. Conclusion et Perspectives

Dans cet article, nous avons proposé une nouvelle approche pour la conception d'un serveur d'information de voyageur fondée sur le paradigme agent et sur l'approche EASI. L'idée est de permettre à un usager de paramétrer ses besoins en information suivant son contexte en générant et en modifiant dynamiquement ses filtres sur la masse d'information disponible sur le serveur.

Nous avons axé notre présentation sur l'échange de flux d'information entre les acteurs du système. L'un des aspects les plus importants est la gestion de l'hétérogénéité des informations mais aussi celle des services disponibles sur le serveur. Il serait donc intéressant de tester l'implantation d'un tel système dans différents réseaux avec différents services afin de vérifier son niveau de flexibilité et de transparence vis à vis des usagers. Hormis le domaine d'application du transport, notre serveur d'information est propice à la formation de coalitions entre applications informatiques offrant un service d'information à un utilisateur final. Les autres perspectives relatives à ce travail concernent les apports qu'ont les services externes quant à la qualité de la palette de services qu'offre le serveur. Nos travaux futurs se focalisent particulièrement sur le service de transport à la demande.

### 6 Bibliographie


Balbo F., ESAC: un Modèle d'Interaction Multi-Agent utilisant l'Environnement comme Support Actif de Communication. Application à la gestion des Transports Urbains, Thèse de doctorat, Université Paris IX Dauphine, Janvier 2000.

Balbo F., Pinson S., Towards a Multi-Agent Modelling Approach for Urban Public Transportation Systems, dans Omicini A., Petta P., Tolksdorf R. Eds., *Engineering Societies in the Agents Worlds II*, LNAI 2203, Springer Verlag, Prague, 2001.

Bichler, M. "A Roadmap to Auction-based Negotiation Protocols for Electronic Commerce", Proceedings of the *33rd Annual Hawaii International Conference on Systems Science*, HICCS, Hawaii, 2000.

Decker K., Sycara K., Williamson M., "Middle-Agent for the Internet", *Fifteenth International Joint Conference on Artificial Intelligence*, 578-583, Morgane Kaufmann, 1997.

De Loor P., Septseault C., Chevaillier P., "Les émotions : Une métaphore pour la résolution de problèmes dynamiques distribués », Briot J.-P., Ghédira K., editors, *Déploiement des systèmes multi-agents, vers un passage à l'échelle*, JFSMA'2003, Revue des sciences et technologies de l'information, volume hors série, pages 331-344. Hermes, November 2003.

Dugdale J.; Pavard B., Soubie J.L., "A pragmatic development of a computer simulation of an emergency call center". In *Designing Cooperative Systems. Frontiers in Artificial Intelligence and Applications*, Rose Dieng et al., IOS pres 2000.

Ferber J., *Les Systèmes Multi-Agents : vers une intelligence collective*, InterEdition, 1995

Ferber J., Reactive distributed artificial intelligence: principles and applications, in G. O'Hare, N. Jennings (Eds), *Foundation of Distributed Artificial Intelligence*, Wiley, New York , 1995b

Finin T., Fritzon R., McKay D., McEntire R., “KQML as Agent Communication Language”, *CIKM’94*, ACM Press, 1994.

FIPA ACL 99 Specification Part2, "Agent Communication Language", www.fipa.org

JAXM 02, « Java API for XML Messaging », juin 2002, Sun Microsystems, http://java.sun.com.

Knoblock C.A., Arens Y., Hsu C. N., “Cooperating Agents for Information Retrieval”, In proceedings *of the Second International Conference on Cooperative Information Systems*, University of Toronto Press, Canada 1994.

O’Brien P. D., Nicol R. C., ‘FIPA — towards a standard for software agents’, BT Technol J, 16, 3, pp 51—59, July 1998.

O’Sullivan D., Nùnez-Suàrez J., Brochoud H., Cros P., Moore C. and Byrne C., “Experiences in the use of FIPA agent technologies for the development of a Personal Travel Application”, Dans les actes de *International Conference on Autonomous Agents (AGENTS)*, Barcelone Septembre 2000.

Petit-Roze C., Grislin-Lestrugeon E., Système d’information à base d’agents, R. Mandiau, E. G. le strugeon, and A. Peninou. *Organisation et applications des SMA*. Hermès, 2002

Predit, Publication Actions fédératives Intermodalité-Voyageurs / Information-communication Rapport final du groupe de définition

http://www.predit.prd.fr/02%2Dpredit/01/publication/fiches/pub0043.htm

Salembier P., "Assistance coopérative aux activités complexes : l’exemple de la régulation du trafic aérien". Dans *Systèmes Coopératifs : de la modélisation à la conception*, In B. Pavard (ed.), Octares, Toulouse, 1994.

Schleiffer R., Transportation Research part C : emerging technologies (special issue), Volume 10C, Numbers 5-6: Intelligent agents in traffic and transportation, 202 pages, 2002

SOAP 01 «SOAP Version 1.2 », 17 Décembre 2001, http://www.w3.org/TR/SOAP.