



HAL
open science

Architecture à base d'agents pour optimiser les services d'aide à la mobilité : vers une conception d'un espace ubiquitaire

Ayoub Bouselmi, Hayfa Zgaya, Thomas Bourdeaud'Huy, Slim Hammadi

► To cite this version:

Ayoub Bouselmi, Hayfa Zgaya, Thomas Bourdeaud'Huy, Slim Hammadi. Architecture à base d'agents pour optimiser les services d'aide à la mobilité : vers une conception d'un espace ubiquitaire. Majec-STIC 2012, Oct 2012, Lille, France. hal-00755750

HAL Id: hal-00755750

<https://hal.science/hal-00755750>

Submitted on 21 May 2013

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.

MajecSTIC 2012
Lille, France, du 29 au 31 octobre 2012

Architecture à base d'agents pour optimiser les services d'aide à la mobilité : vers une conception d'un espace ubiquitaire

Ayoub Bousselmi¹, Hayfa Zgaya², Thomas Bourdeaud'huy¹, Slim Hammadi¹

1 : Ecole Centrale de Lille, Cité Scientifique - BP 48, 59651 Villeneuve D'Ascq cedex - France.

2 : Université Lille 2 Droit et Santé, Institut Lillois d'Ingénierie de la Santé, 42 rue Ambroise Paré, 59120 LOOS - France.

Contact : {ayoub.bousselmi, thomas.bourdeaud_huy, slim.hammadi}@ec-lille.fr, hayfa.zgaya@univ-lille2.fr

Résumé

Les entités de l'Intelligence Artificielle, communicantes avec leur entourage, telles que les PDAs, les téléphones, les capteurs, les robots, les logiciels, les middlewares, etc. sont de plus en plus présentes dans notre environnement. La nouvelle topographie de nos espaces quotidiens introduit les notions de l'Intelligence Artificielle Distribuée, l'Intelligence Ambiante et l'Informatique Ubiquitaire. Notre objectif est d'avoir un accès en temps réel, via un support mobile, à différents types d'information issus d'un environnement ubiquitaire, tels que les lieux touristiques et les horaires de bus. Dans ce papier, nous proposons une architecture efficiente d'un système ubiquitaire ciblé basé sur une approche multi-agent. Nous nous focalisons dans notre étude sur le Grand Stade Lille Métropole et les services qui peuvent y être demandés. L'architecture proposée va devoir supporter les services et optimiser leur qualité par rapport au temps de réponse et au coût de l'information.

Abstract

Artificial Intelligence entities, capable to communicate with their vicinity, such as PDAs, Smartphones, sensors, robots, software, middleware, etc. are increasingly introduced in our environment. The conventional topography of our everyday space introduces Distributed Artificial Intelligence, Ambient Intelligence and Ubiquitous Computing concepts. For example, when traveling to a new city, information about sights in the city, hotels, or even bus timetable are needed. Thanks to ubiquitous computing, citizens as well as tourists can communicate with their environment and get all information in real time in their terminals. In this paper, we present a large set of ubiquitous services that might be requested in a ubiquitous space (in this case, the ubiquitous space is the Grand Stade Lille Métropole) and we design a multi-agents architecture which will support these services and optimize their quality such as response time and information cost.

Mots-clés : Intelligence Artificielle Distribuée, Espace Ubiquitaire, Services Ubiquitaires, Architecture Multi-Agent.

Keywords: Distributed Artificial Intelligence, Ubiquitous Space, Ubiquitous Services, Multi-Agent Architecture.

1. Introduction

Partant de la capacité des systèmes multi-agents à bien modéliser les problèmes distribués, nous nous sommes focalisés sur le problème d'optimisation des services d'aide à la mobilité

dans les espaces ubiquitaires. Dans le cadre d'un projet gagné par le LAGIS¹ avec OSEO² et l'entreprise ARIZONA une étude fonctionnelle du Grand Stade Lille Métropole³ a été menée. Notre objectif est de concevoir une suite servicielle pour l'aide à la mobilité ainsi que l'architecture informatique qui va les supporter. Il n'est pas uniquement un stade de foot car il peut aussi se transformer en une salle de concert. Notre objectif est de retenir les personnes présentes le plus longtemps possible. Par exemple, le match peut durer deux heures et donc le but est d'attirer ces personnes une heure avant, et de les retenir au moins une heure après en assurant leur confort et divertissement, d'où le besoin d'un ensemble de services d'accompagnement du public. Dans ce contexte, cet espace ubiquitaire, comportera des services en ligne, des restaurants, des magasins, des hôtels [4], des jeux et même des crèches pour les enfants. La demande de ces services dans un tel espace ubiquitaire sera massive, donc l'architecture informatique doit nécessairement optimiser la qualité de ces services en matière de temps de réponse et de coût de l'information. Dans ce papier, nous proposons une architecture distribuée basée sur un système multi-agents qui permettra de supporter un ensemble de services ubiquitaires. Le reste de ce papier est organisé comme suit : la section 2 présente l'état de l'art ; la section 3 illustre les services ubiquitaires suggérés ; la section 4 décrit l'architecture multi-agents proposée ; la section 5 illustre une simulation du service U-Fun ; finalement, la section 6 fournit la conclusion et les perspectives de notre travail.

2. Etat de l'Art

Un espace ubiquitaire est un espace qui comporte des ordinateurs et des capteurs intégrés partout pour traiter et fournir des informations, des services et des ressources en temps réel n'importe quand et n'importe où. Selon Weiser [7, 8], l'Informatique Ubiquitaire (traduction du mot anglais *ubiquitous computing*, la forme rétrécie est *ubicomp*) et la Réalité Virtuelle (RV) sont deux paradigmes opposés. En effet, la RV nous amène au sein d'un monde virtuel créé par l'ordinateur, par contre le *ubicomp* amène les ordinateurs à notre monde réel et l'enrichit avec des informations. L'*ubicomp* permet le contrôle à distance, la connexion permanente des individus, l'établissement des relations communautaires en ligne, le travail à distance, etc. Ce concept possède plusieurs nominations : l'informatique pervasive ou omniprésente, l'informatique invisible, l'informatique disparaissante, l'informatique consciente, l'intelligence ambiante et l'informatique calme [2] (traduction respective des termes anglais suivants : *Pervasive computing*, *Invisible computing*, *Disappearing computing*, *Sentient computing*, *Ambient Intelligence (AmI)* et *Calm computing*). Grâce aux nouvelles technologies, les appareils informatiques sont de plus en plus miniaturisés et cachés (les Smartphones, les PDAs, PCs, les moniteurs de fréquences cardiaques HRMs, etc.) et capables d'offrir des services d'aide à la mobilité. Tomas Plötz *et al* [6] proposent une approche de modélisation des espaces intelligents basée sur la modélisation de l'être humain comme étant un élément du système qui peut fournir lui aussi des services afin de surpasser le problème « all-or-nothing ». Alvaro Fides-Valero *et al* [1] proposent la plateforme PERSONA (PERceptive Space prOmoting iNdependent Aging) d'assistance de l'autonomie des personnes âgées. Cette architecture s'inspire de la combinaison des deux modèles SODAPOP et Amigo⁴ (2004-2008, FP6-IST). Elle profite du système de communication développé dans SODAPOP et de la couche servicielle et les solutions d'ontologies de l'architecture Amigo. PERSONA est basée sur une architecture ad-hoc où chaque nœud possède son propre intergiciel qui lui permet de découvrir les autres nœuds et par la suite communiquer et coopérer avec eux. PERSONA propose sept types de services pour les espaces d'assistance ambiante des personnes. Ces travaux sont limités à des petits espaces (une maison, une chambre, une salle de conférences etc.) avec un nombre limité d'interactions utilisateur-système mais jusqu'à présent il n'y a pas d'architectures d'espace ubiquitaire développées à l'échelle d'un stade ou bien d'une ville entière. La ville ubiquitaire pleine d'ordinateurs interconnectés peut être vue comme un environnement distribué où les

¹ <http://lagis.ec-lille.fr/>

² <http://www.oseo.fr/>

³ <http://www.grandstade-lillemetropole.com/>

⁴ <http://cordis.europa.eu/projects/index.cfm>

terminaux peuvent être, au même temps, des chercheurs ou bien des fournisseurs d'information. Les architectures traditionnelles centralisées ne peuvent pas répondre à notre objectif puisqu'il y aura une multitudes de services ainsi qu'un nombre d'utilisateurs très grand. Donc on a besoin d'une architecture moderne qui soit à la fois robuste, flexible et extensible pour gérer le flux de données et optimiser des critères bien déterminés. Une architecture distribuée, basée sur un système multi-agents est convenable pour ce type d'environnement. Une telle architecture a les avantages d'être enfichable, étendue et susceptible d'intégrer différents algorithmes d'optimisation. En effet, les algorithmes d'optimisation montés dans une architecture distribuée à base d'agents (que nous les appellerons agents optimisateurs) ont deux principaux rôles : l'instauration de la qualité de service requise ainsi que la comparaison des différents algorithmes pour décider lequel est le plus adapté à chaque situation.

3. Services Ubiquitaires Conçus

Nous avons proposé plusieurs services qui aident à la mobilité des personnes, qui assurent le divertissement et préservent la sécurité dans un territoire ubiquitaire. En effet, cinq grands thèmes de services sont identifiés: Services dans le stade, Service pour le transport, Services touristiques, Services culturels, Services de sécurité.

3.1. Services dans le Stade

Ces services sont identifiés pour le stade. Ils comportent des services d'aide à la mobilité ainsi que des services de divertissement. Cette liste n'est pas exhaustive et il peut y avoir d'autres services. (La lettre U est un acronyme du mot Ubiquitous).

Service U-Fun: Ce service permet d'organiser des chorégraphies avant et au cours des spectacles. Il permet d'organiser des messages géants au cours des matchs en fonction des places dans les gradins. Au cours d'un concert aussi, il est possible de faire des chorégraphies avec la lumière des écrans de Smartphones en créant des alchimies et des mouvements de lumière ainsi que le suivi des paroles des chansons (i.e. mode karaoké).

Service U-Goal : Ce service propose à l'utilisateur, lors du visionnement d'un but ou d'une occasion de but, de changer l'angle de vue en choisissant une autre caméra (pour se rassurer des décisions de l'arbitrage par exemple). Il permet aussi aux supporters de regarder les buts et les occasions de tous les matchs qui se jouent en même temps.

Service U-Player : Ce service regroupe toutes les informations des célébrités (joueurs, chanteurs, comédiens, humoristes, etc.). Il permet d'avoir une fiche pour chaque joueur : statistiques de buts, cartons, etc. télécharger ou voir en streaming les meilleurs buts de l'équipe ou les meilleurs vidéo-clips des musiciens, et avoir les statistiques du match en temps réel. Ce service offre également à l'utilisateur les résultats en temps réel des matchs qui se jouent en parallèles ainsi que les statistiques (Possession de balle, cartons, tirs cadrés, buts, etc.).

Service U-BabySitter : Sachant que le stade peut offrir la prestation de garde d'enfants. Ce service permet de surveiller à distance ces enfants qui se trouvent dans une garderie ou une salle de jeux dans le stade. Ce service permet de: Géolocalisation les enfants dans le stade (accès strictement privé) et pouvoir surveiller les enfants en direct sur le Smartphone et parler à des baby-Sitters en ligne au besoin.

3.2. Services de Transport

Pour faciliter le déplacement depuis et vers le stade, nous avons identifié les deux services : U-Parking et U-GPS+.

Service U-Parking: Ce service permet de: Montrer le nombre de places libres dans un parking, aider le conducteur à trouver sa place dans le parking et détecter les prix des parkings.

Service U-GPS+ : Ce service permet aux personnes à mobilité réduite de pouvoir circuler dans le stade ou dans les alentours grâce à un GPS vocal et visuel qui les guidera vers l'endroit souhaité.

3.3. Services Touristiques

Tout au long du voyage vers le stade, l'utilisateur peut profiter des services suivants :

Service U-Map : Ce service sera utilisé entre amis, membres d'une famille, groupes de touristes, etc. Il permet le partage en temps réel de la position des adhérents de l'application pour faciliter leurs rencontres et donne le sentiment de la promenade ensemble.

Service U-Meal : Ce service permet de réserver une table dans un restaurant, de choisir un menu (en choisissant la langue ce qui facilite l'usage par les touristes), trouver facilement la table réservée, commander et commenter la qualité des services et des repas.

Service U-Travel: Ce service permet d'acheter en ligne des tickets numériques, jouer à des jeux multi-joueurs, partager des photos, accéder à une bibliothèque en ligne (livres électroniques, morceaux de musique, films), suggérer des plans touristiques et noter la qualité des services.

3.4. Services Culturels

Nous avons identifié deux services U-Culture et U-Book dans le domaine culturel.

Service U-Culture: Ce service permet de fournir des informations multimédia sur les monuments de la ville et leur histoire.

Service U-Book: Ce service représente une bibliothèque numérique en ligne, (concept de la bibliothèque municipale), qui permet de choisir les livres suivant une recherche par le nom, l'ISBN, ou l'auteur par exemple. Il peut connaître les préférences des lecteurs et leurs proposer d'une manière intelligente des livres pouvant les intéresser.

3.5. Service de Sécurité : U-Security

Ce service permet d'aider les policiers et les pompiers en géolocalisant l'appel de secours et de compter le flux des personnes dans le stade ou dans les transports en commun en temps réel.

Ceci permet d'organiser les personnes dans les portes de sorties et cas d'incendie, connaître le nombre de supporters pour organiser les stadiers, faire des statistiques des gradins et des portes les plus empruntées, etc. Il est très important d'identifier les heures de pointe avec précision dans les transports en commun afin d'adapter le nombre de véhicules selon le nombre de passagers et améliorer la qualité des services.

3.6. Analyse des services

Le Grand Stade Lille Métropole, comme indiqué plus haut, peut se transformer en une aréna pour organiser des concerts. Pendant les matches de foot ou de rugby, les personnes vont demander le replay du but par exemple à l'aide du service U-Goal. Ainsi, 50.000 personnes sont susceptibles de demander simultanément le même service et l'architecture informatique du stade doit pouvoir gérer ce flux important de requêtes. De même, dans le stade lorsqu'il est dans sa forme d'Aréna, 30.000 personnes peuvent demander le service U-Fun par exemple, pour télécharger les paroles de la chanson en cours sur leurs terminaux. Ceci doit s'effectuer rapidement et par conséquent, l'architecture informatique dans le stade, en plus de la gestion de la montée en charge elle doit optimiser les temps de réponse des services. Au contraire, la sollicitation d'autres services tels que U-Book, U-Parking, U-Meal et U-Travel est un peu moins importante. Il faudra que l'architecture informatique adoptée optimise le coût des services. Par exemple, plusieurs parkings proposeront leurs services à des coûts différents à des emplacements différents (par rapport au stade), et de ce fait, l'affectation des stadiers motorisés au parkings doit nécessairement être optimisée.

Trois contraintes doivent être bien gérées par l'architecture adoptée : La demande massive des services, l'optimisation du coût des services, l'optimisation du temps de réponse.

Les architectures classiques centralisées sont convenables pour des services à petite échelle. Par contre, pour les jeux multi-joueurs par exemple, un seul serveur est adéquat lorsque le nombre de joueurs simultanés est de l'ordre d'une centaine de joueurs mais il ne peut gérer des milliers de joueurs simultanés [5]. D'autre part, vouloir centraliser la gestion du flux de requêtes sur un seul serveur pour l'optimisation des critères risque de retarder les réponses du système. De ce fait, une architecture distribuée orientée-agents est propice pour gérer le flux massif de requêtes utilisateurs et optimiser le coût des services et le temps de réponse. Nous présentons cette architecture dans la section suivante.

4. Architecture Multi-Agents

4.1. Les Système Multi-Agents

Les systèmes multi-agent (SMA) ont été un sujet de recherche depuis plusieurs années et ils le sont encore aujourd'hui de la demande croissante des solutions informatiques pour des problèmes complexes et distribués. Un SMA est composé de plusieurs entités logicielles ou bien matérielles appelées des agents. Ces entités ne sont pas nécessairement complexes et possèdent plusieurs caractéristiques et capacités qui leurs permettent de vivre ensemble et gérer des problèmes complexes par un travail d'équipe. Selon Jennings et Wooldridge [3] il y a quatre principales caractéristiques communes entre les agents :

- L'autonomie : les agents sont capables d'exécuter la plupart de leurs tâches sans l'intervention de l'être humain ou d'autres agents et possèdent un certain degré de contrôle de leurs actions et de leurs états internes.
- La sociabilité : les agents sont capables d'interagir au bon moment avec d'autres agents ou êtres humains pour compléter leurs tâches ou aider les autres agents.
- La réactivité : Les agents doivent comprendre et apercevoir leur environnement et répondre aux changements qui s'y produisent.
- La pro-activité : En plus de répondre aux stimulations de l'environnement, les agents doivent être opportunistes, avoir un comportement dirigé par un objectif et prendre l'initiative quand il le faut.

Un agent, situé dans un environnement réel ou virtuel, peut jouer un ou plusieurs rôles dans une organisation, est motivé par des tendances internes et cherche à satisfaire ses propres objectifs. L'agent peut être capable de communiquer avec d'autres agents. Il peut agir sous la conduite d'un autre utilisateur ou d'une autre entité, anticiper, apprendre de ses expériences et s'adapter à son environnement. Dans un SMA, les agents opèrent d'une manière décentralisée. Ils coordonnent leurs actions afin d'éviter les conflits de ressources pour atteindre collectivement, le même objectif (des agents non antagonistes), ou des objectifs divergents (des agents négocient et nous parlons d'agents antagonistes). Les SMA peuvent être utilisés de deux manières : pour simplifier la description d'un problème en distribuant les tâches et les informations ou bien pour définir des modèles de simulation. Dans ce cas les agents sont dotés d'un comportement et évoluent dans un environnement pour étudier le comportement global du système.

4.2. Vue Globale du Réseau

Les informations nécessaires pour répondre à des requêtes utilisateur sont de plus en plus hétérogènes et distribuées sur des réseaux à grande échelle. De ce fait, la sélection de la source adéquate pour chaque information ainsi que la gestion d'un nombre important de sollicitations simultanées est un défi perdu pour une architecture centralisée. Pour parvenir à satisfaire les besoins introduits par des services ubiquitaires, nous proposons la Plateforme de REcherche et de Composition des Services d'Aide à la Mobilité (PRECSAM), basée sur un système multi-agents (Figure 1).

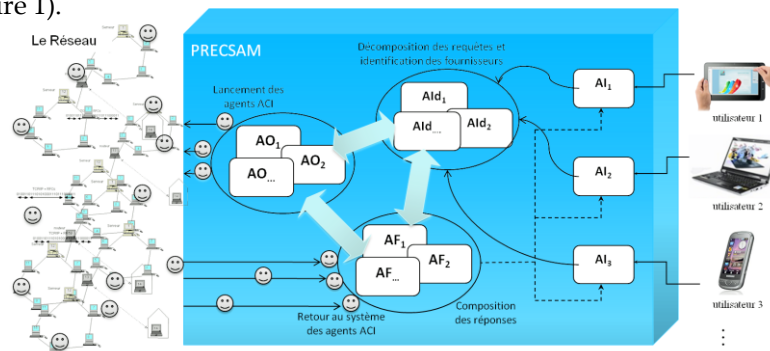


Figure 1 - Vue globale de l'architecture

L'objectif global de notre travail est de mettre en œuvre une PRECSAM dans un espace

ubiquitaire. Cette plateforme sera capable d'être sollicitée n'importe quand et de n'importe quel endroit et de gérer un nombre important de requêtes utilisateur qui peuvent être simultanées à partir de n'importe quel support fixe ou mobile. La PRECSAM doit être responsable du traitement des requêtes utilisateurs, de la recherche des informations nécessaire sur le réseau et de l'optimisation du temps de réponse et du coût de l'information. Nous avons défini cinq types d'agents qui vont correspondre au noyau de notre système.

- Des agents interfaces (AI) : le moyen de communication avec l'utilisateur des services.
- Des agents identificateurs (AId) : pour récupérer les blocs de requêtes, les décomposer et ensuite identifier les similitudes et les fournisseurs d'informations nécessaires.
- Des agents optimisateurs (AO) : pour optimiser les plans de routes des agents collecteurs d'informations (agents mobiles).
- Des agents mobiles (AM) : ayant la capacité de se déplacer d'un nœud à un autre sur le réseau afin de scruter les informations nécessaires pour répondre aux requêtes utilisateurs.
- Des agents de fusion (AF) : pour récupérer les données collectées, les composer et les envoyer aux agents interfaces correspondants pour répondre aux utilisateurs.

Ces agents sont communicants, avec l'utilisateur, entre eux et avec le reste du réseau. Nous détaillerons dans ce qui suit la description de la problématique et nous expliquerons le fonctionnement dynamique du système.

4.3. Description du Problème

Une requête client doit être décomposée et la réponse sera la composition d'un ensemble de données hétérogènes provenant de plusieurs fournisseurs. Lorsque nous considérons un très grand nombre de requêtes utilisateurs nous mettons en évidence le problème du temps de réponse qui doit être le plus court possible (nous l'appellerons période d'acquisition et la noterons par Δ_ϵ). Pour cela, dans un premier niveau d'optimisation, le système doit d'abord décomposer l'ensemble des requêtes acquises en sous requêtes que nous appelons tâches, en identifiant les similarités. Pendant cette étape, l'agent AId rassemble les blocs similaires et fait la demande de chaque service une seule fois pour optimiser les flux de données.

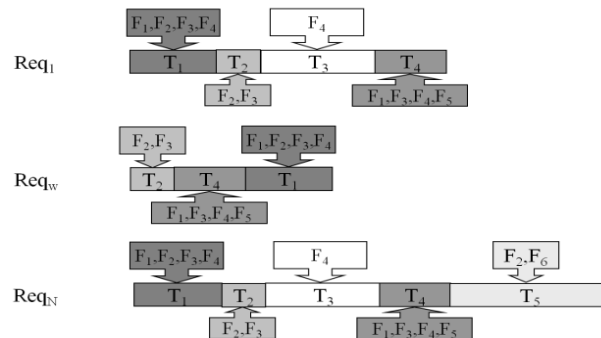


Figure 2 [9]- Identification des nœuds

Chaque service sollicité peut être proposé par plusieurs fournisseurs ($F_1, F_2, F_3, F_4, \dots$) avec différentes dates d'échéance (d_i), différents temps de réponse, coûts et tailles des données (Figure 2). Nous sommes donc face à un problème multi-objectif qui vise à trouver la meilleure affectation des serveurs fournisseurs aux services demandés en cherchant les meilleurs temps de réponse et coût de service disponibles. Lorsqu'il y a des blocs similaires entre les différentes requêtes utilisateurs l'AID détecte les similarités entre ces requêtes ce qui va diminuer éventuellement le temps de réponse.

4.4. Comportement Dynamique du Système

A l'état initial, le système au repos, les ressources ne sont pas utilisées puisque les agents ne sont pas encore créés. En effet, le cycle de vie d'un agent dépend de l'arrivée des requêtes utilisateurs. Dès l'arrivée d'un ensemble de requêtes pendant la période d'acquisition Δ_ϵ (comptée à partir de l'arrivée de la première requête à $t=t_1$), une société d'agents (P_{t1}) est créée pour commencer le traitement de ces requêtes (Figures 3).

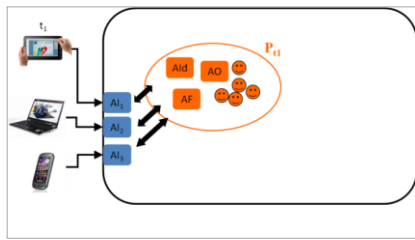


Figure 3 - Création de la première société d'agents

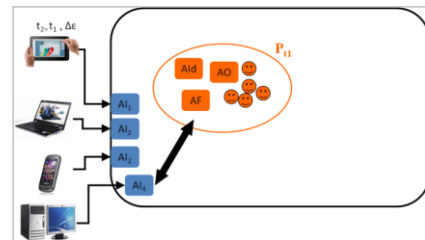


Figure 4 - Nouveau cycle de traitement des requêtes utilisateurs

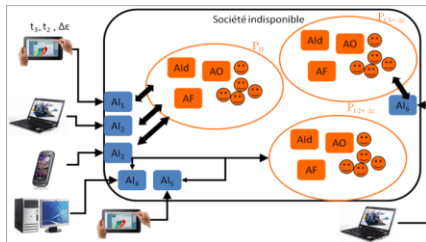


Figure 5 - Création d'une nouvelle société

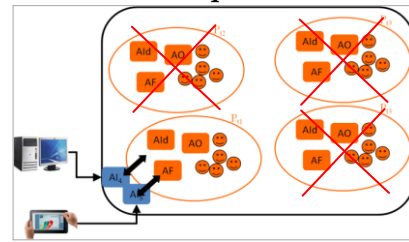


Figure 6 - Destruction des sociétés en état d'inactivité

Après avoir servi les demandes précédentes, la société est prête à traiter de nouvelles requêtes utilisateurs (à partir de $t=t_2=t_1+\Delta\epsilon$), (Figure 4), et le même scénario se répète. Nous appelons les agents AI_d, AF, AO et AM ainsi créés à l'instant « t », la société d'agents P_t. Ensuite, et pendant une courte période de temps, que nous appelons période d'acquisition et notons par $\Delta\epsilon$, chaque nouvelle requête formulée par un nouvel agent AI, est gérée par la même société d'agents P_t dont la création a été déclenchée par la première requête formulée à l'instant « t ». Si à l'instant « t+ $\Delta\epsilon$ », toutes les sociétés d'agents préalablement créées sont indisponibles, alors la formulation de nouvelles requêtes déclenche la création d'une nouvelle société P_{t+ $\Delta\epsilon$} et ainsi de suite (Figure 5). Dès qu'une société P_t créée à l'instant « t », est disponible, elle devient prête à gérer un ensemble de requêtes formulées pendant une nouvelle période $\Delta\epsilon$ à un nouvel instant « t_j » avec $t_i \leq t_j$, nous appelons les requêtes formulées pendant la période $\Delta\epsilon$, des requêtes $\Delta\epsilon$ – simultanées. Cependant, dès que la disponibilité d'une société ou d'un agent AI atteint la période $\Delta\infty$, alors cette société ou cet agent est automatiquement détruit (Figure 6).

5. Simulation du Service U-Fun

Le service U-Fun permet à son utilisateur de récupérer la couleur de la pancarte qu'il doit porter avec lui dans le stade, ce qui permet de faire des grandes chorégraphies dans le stade et écrire des messages géants. Dans cette démonstration, développée à l'aide de JADE⁵, deux agents sont utilisées : AI, AO. La figure 7 montre l'outil graphique RMA de la plateforme JADE et la figure 8 montre le flux de requêtes entre ces deux agents.

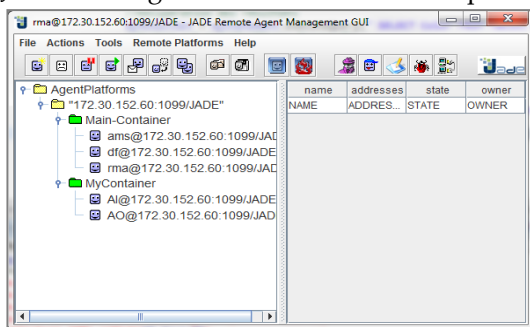


Figure 7 - Plateforme du système

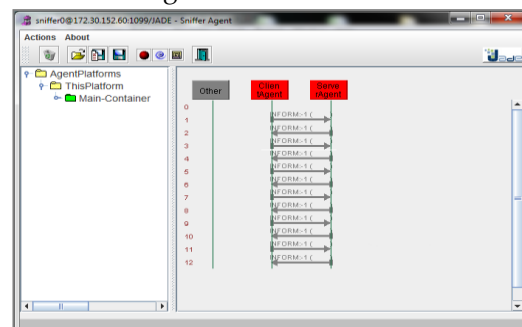


Figure 8 - Communication intra-agent

⁵ <http://jade.tilab.com/>

- AI : c'est l'agent responsable de la récupération de la demande utilisateur, la formulation et l'envoi de la requête au AO ainsi que l'affichage du résultat.
- AO : c'est l'agent responsable du traitement de la requête utilisateur.

Cette démonstration n'utilise que deux agents AI et AO puisqu'ils suffisent pour traiter les demandes utilisateurs. En effet, le test est réalisé en local, donc nous n'avons pas besoin de l'AM, ni des agents AId et AF (la requête est simple, pas besoin d'identifier les similarités). Elle nous a permis de valider initialement par ce service élémentaire l'architecture distribuée à base d'agents. Les simulations seront plus développées dans les prochains travaux.

6. Conclusion et perspectives

La complexité et le caractère distribué du problème de conception d'une architecture informatique d'un espace ubiquitaire nous ont ramené à l'adoption d'une approche multi-agent pour la conception d'une architecture orientée-services tout en optimisant le temps de réponse ainsi que le coût des informations. Les services ubiquitaires introduits nous ont permis de fixer les critères à optimiser dans l'espace ubiquitaire concerné et ont influencé la conception de l'architecture. Nous allons intégrer dans les prochains travaux plusieurs types d'algorithmes d'optimisation au sein des agents et les faire fonctionner en mode concurrence pour étudier l'efficacité des algorithmes selon les différentes contraintes.

Bibliographie

1. Álvaro Fides-Valero, Matteo Freddi, Francesco Furfari, Mohammad-Reza Tazari; "The PERSONA Framework for Supporting Context-Awareness in Open Distributed Systems", AmI '08 Proceedings of the European Conference on Ambient Intelligence.
2. Frank Stajano; "Security Issues in Ubiquitous Computing" Handbook of Ambient Intelligence and Smart Environments, 2010.
3. Jennings, N.; Wooldridge, M.; "Software agents," IEE Review, vol.42, no.1, pp.17-20, 18 Jan 1996.
4. Le Grand Stade Lille Métropole (<http://www.grandstade-lillemetropole.com/>)
5. Merabti, M.; El Rhalibi, A.; , "Peer-to-peer architecture and protocol for a massively multiplayer online game," Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE , vol., no., pp. 519- 528, 29 Nov.-3 Dec. 2004
6. Thomas Plötz, Christian Kleine-Cosack, Gernot A. Fink; "Towards Human Centered Ambient Intelligence", AmI '08 Proceedings of the European Conference on Ambient Intelligence.
7. Weiser, M.; "Hot topics-ubiquitous computing" Computer, vol.26, no.10, pp.71-72, Oct 1993.
8. Want, R.; Schilit, B.N.; Adams, N.I.; Gold, R.; Petersen, K.; Goldberg, D.; Ellis, J.R.; Weiser, M.; "An overview of the PARCTAB ubiquitous computing experiment," Personal Communications, IEEE , vol.2, no.6, pp.28-43, Dec 1995
9. Zgaya, H.; Hammadi, S.; , "Assignment and Integration of Distributed Transport Services in Agent-Based Architecture," Intelligent Agent Technology, 2006. IEEE/WIC/ACM International Conference on , vol., no., pp.96-102, 18-22 Dec. 2006