

Description en LDS des services basés sur la localisation

Amel Mederreg

Institut National des Télécommunications
9, rue Charles Fourier
91011 Évry cedex, France
Amel.Mederreg@int-evry.fr

Ana Cavalli

Institut National des Telecommunications
9, rue Charles Fourier
91011 Évry cedex, France
Ana.Cavalli@int-evry.fr

Résumé

Dans cet article, nous présentons le fonctionnement ainsi que la description formelle en LDS des services basés sur la localisation appliqués dans le domaine de la téléphonie mobile, et qui sont étudiés dans le cadre du projet PLATONIS. Ces services permettent d'accéder à de nouveaux services ciblés sur la position en temps réel du demandeur (météo, itinéraire, etc.). À partir de la description formelle de ces services, un ensemble de scénarios a été généré automatiquement, en utilisant un algorithme pour le test de composants. Cet ensemble de scénarios va permettre de vérifier le comportement des services par rapport à leurs descriptions formelles.

Mots clef : Service de localisation, Wireless Application Protocol, Langage de Description et Spécification, test de composants.

1 Introduction

Les réseaux mobiles et notamment la téléphonie cellulaire comme le système GSM (Global System for Mobile communication) sont une des avancées majeures de cette dernière décennie. Le nombre d'abonnés ne cesse de croître et va bientôt rejoindre celui du réseau téléphonique classique. Parallèlement à la téléphonie cellulaire, les ordinateurs portables et les assistants numériques personnels ont connu eux aussi un développement très rapide. Nous nous dirigeons donc vers un monde où les appareils portables vont être légions et seront reliés entre eux et avec des serveurs fixes par moyen de communication sans fil.

Cette tendance, en plus d'utiliser les protocoles et services qui existent actuellement sur Internet et la téléphonie classique, a donné naissance à de nouveaux protocoles et services. Parmi eux, on retrouve les protocoles de la pile WAP (Wireless Application Protocol) qui a été conçu pour permettre la transmission et la présentation de documents Internet sur des périphériques dont les possibilités d'affichage sont limitées.

Dans la plupart des cas, ces nouveaux services et protocoles devront mettre en relations des éléments hétérogènes, qu'il faudra tester pour garantir leur interoperabilité ; pour cela la réalisation d'une spécification formelle avant toute implantation s'avère nécessaire. Dans ce cadre, un projet appelé PLATONIS (PLAte-forme de validaTiOn et d'experimentaion multi-protocoles et multi-Services), a été introduit par le RNRT, en collaboration avec des laboratoires de recherche et des groupes industriels [6]. Ce projet a pour objectif l'étude, sur une plate-forme de test, des protocoles et services liés à la mobilité et des technologies permettant son développement. Deux types de tests seront exécutés : le test d'interopérabilité et le test de conformité. Le test d'interopérabilité a été conçu pour la validation et l'expérimentation des nouveaux services liés à la mobilité. Le test de conformité a été conçu pour vérifier que l'implantation du protocole est faite selon la spécification standard.

Parmi les services usités pour les tests d'interopérabilité entre une application sur terminal et une application sur un serveur, on retrouve dans le cadre du projet PLATONIS, des services qui utilisent la fonction de localisation. Cette fonction permet aux services d'acheminer une information et de la mettre à la disposition de l'utilisateur en se basant sur son emplacement à la minute près.

Dans cet article, nous présentons les techniques utilisées pour la localisation ainsi que la spécification formelle et la simulation d'un service de localisation d'abonné. Nous présentons également, un ensemble de scénarii générés à partir de la description formelle et qui permettent de tester que différentes implantations sont conformes à la spécification.

L'article est organisé comme suit : La section 2 introduit l'architecture et la technologie liées à la fonction de positionnement. La section 3 décrit les services choisis par le projet PLATONIS et qui utilisent cette fonction. La section 4 détaille la spécification en LDS (Langage de Description et Spécification) des services et relate, avec les séquences de test MSC obtenues après utilisation d'un algorithme de test imbriqué. Enfin, la section 5 conclut l'article.

2 Fonction de localisation

LoCation Services (LCS), appelé aussi Mobile Positioning, est un ensemble de services dépendants de la connaissance de la position géographique du terminal. La position du terminal est une information qui peut être demandée par une application, à qui on attribue la fonction "client". L'application peut appartenir au réseau, lui être étrangère ou exister seulement sur le terminal. La requête est reçue par le serveur (MPC : Mobile Positioning Center) qui la traite et la renvoie à l'application demandeuse. Le MPC récupère l'information de la localisation moyennant l'une des technologies de positionnement suivantes : CGI+TA, E-OTD, UL-TOA, A-GPS...etc.

Le Client et le Serveur de localisation communiquent entre eux en utilisant le protocole MPP (Mobile Positioning Protocol). Ce dernier est un protocole basé sur HTTP ce qui permet l'accès aux informations liées à la localisation sur n'importe quelle plate-forme (figure 1) [7, 1, 11].

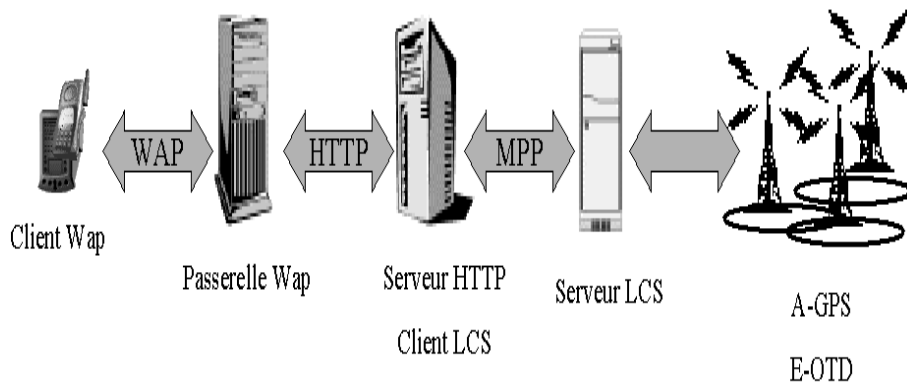


FIG. 1 - Exemple de procédure de demande de localisation

Requête et réponse MPP :

- Le format de la requête est l'URL suivante : `http://position.on.demand/PositionRequest/Direct?USERNAME=TheUser&PASSWORD=TheUserPassword&POSITION_ITEM=12345678&POSITION_TIME=(timenow)`
- Le résultat de la requête peut être positif ou négatif selon qu'elle soit correctement écrite et autorisée. La réponse peut contenir la position de plusieurs MS (Mobile station)¹, du fait que la requête est faite pour un ou plusieurs terminaux en même temps. Pour dissocier les MS un numéro d'identification est utilisé [7].

1. MS est un terminal mobile: le téléphone portable, un PDA...

2.1 Système de positionnement mobile

Les technologies de positionnement actuelles qu'utilisent les serveurs de localisation, peuvent être divisées en deux groupes : [2]

2.1.1 Les solutions orientées terminal

1. GPS Global Positioning System

Le GPS utilise un ensemble de satellites (entre 3 et 4) pour localiser l'utilisateur. Cette information brute est traitée par le terminal ou envoyée sur le réseau pour être traitée afin de récupérer la position courante. L'exactitude de la position est de 5 à 40m en ciel clair.

2. A-GPS Assisted GPS

Le A-GPS utilise des récepteurs fixes GPS mis (au sol) à des intervalles réguliers entre 200 et 400km, afin d'augmenter la précision de la position et de diminuer le temps de détection.

3. E-OTD Enhanced Observed Time Difference

Le E-OTD utilise un logiciel, résidant sur le terminal et qui utilise les données envoyées par les stations de base et calcule le temps écoulé pour définir sa position. Le calcul peut être effectué par le terminal ou par le réseau. L'exactitude de la position est de 125m, mais à la différence du GPS cette méthode ne dépend pas de la clarté du ciel.

2.1.2 Les solutions orientées réseau

1. CGI-TA Cell Global Identity-Timind Advance

CGI-TA utilise l'identité de chaque cellule pour localiser l'utilisateur. Cette technologie est très peu coûteuse car elle utilise les terminaux déjà existants et peut être déjà utilisée pour les services comme la proximité. La précision varie entre 10 à 500m selon la taille de la cellule.

2. TAO Time Of Arrival

Cette technique est semblable à E-OTD sauf qu'elle prend en compte la taille des données échangées entre le terminal et les stations de base.

3 Application de services liés à la localisation

Comme on l'a mentionné précédemment la clé du succès pour propulser la téléphonie mobile par rapport à l'internet, réside dans les applications capables de prendre en compte la localisation de l'abonné.

Ce type de services fonctionne selon le modèle d'architecture suivant : l'abonné fait une demande de service au serveur, ce dernier une fois qu'il a identifié le type de service, envoie une requête de localisation au serveur MPC pour récupérer la position de l'abonné. Une fois qu'il a obtenu une réponse, il s'en sert pour sélectionner, dans la liste existante, la donnée correspondante à cette position et envoie le résultat au terminal.

On retrouve parmi l'ensemble des services se développant autour de ce concept : [3, 9]

- Le service d'Itinéraire : Ce service aide les abonnés à se déplacer en leur décrivant au mieux leur parcours et en les informant de la distance à parcourir et le temps nécessaire pour le faire [4]. Ce service donne trois types de possibilité pour récupérer cette information :

1. Rue par rue, pour des déplacements à pied, en voiture...
2. Appel vers un centre d'appel, qui est capable de fournir à l'abonné cette information soit directement soit en l'envoyant par un message SMS...

3. Ou juste savoir leur position courante.

Le scénario de communication entre le terminal et le serveur se fait comme suit : le terminal choisit le service d'itinéraire parmi les services proposés dans le menu. Une fois le service choisi, le serveur renvoie les informations concernant le choix que doit faire l'utilisateur sur sa position courante, soit l'appel vers le centre le plus proche, soit choisir d'avoir des informations par rapport à sa destination. Si cette dernière alternative est choisie, l'utilisateur doit communiquer au serveur sa destination. Le serveur une fois qu'il a récupéré ces informations fait une recherche dans sa base de données selon la position courante et envoie le résultat à l'abonné.

- Le service de proximité : Permet d'informer les abonnés des différents points utiles (restaurant, cinéma...) qui se trouvent près d'eux.
Par exemple, si l'utilisateur veut aller au cinéma, alors il choisit dans le menu le service de proximité. Une fois le service choisi, le serveur envoie la liste des points utiles qui se trouvent dans la base de données. Une fois le choix fait par l'utilisateur (pour ce cas le cinéma), le serveur cherche dans sa base de données tous les cinémas qui se trouvent près de l'utilisateur et lui envoie le résultat. L'utilisateur choisit le type du cinéma où il veut aller et envoie la réponse au serveur. Ce dernier récupère l'adresse du cinéma choisi et l'envoie à l'utilisateur. Il donne aussi la possibilité à l'utilisateur de lui indiquer l'itinéraire pour y aller, et donc le service de proximité bascule vers le service itinéraire [4].
- Le service de trafic : Permet de s'informer de l'état du trafic via une carte dédiée ou en appelant un centre de renseignement. L'utilisateur, après avoir reçu la réponse du serveur, choisit entre la carte du trafic où il est et l'appel vers un centre dédié.
- Le service de secours : Permet d'avoir toutes les informations sur les différents centres de secours, qui se trouvent dans les alentours de l'abonné, en fonction de ses besoins. Même procédure que pour le service de proximité.
- Le service de recherche : Permet d'effectuer deux types de services : par rue et par mot-clé. La recherche par rue permet de rechercher une rue et d'y accéder, si elle est connue par le serveur, en utilisant le service itinéraire ou trafic. La recherche par mot-clé permet une recherche parmi tous les points utiles et les points de secours connus par le serveur, et l'accès au résultat comme pour la recherche par rue.

Ces services sont regroupés en une application à laquelle l'utilisateur accède, via son terminal, en choisissant dans un menu (figure 2).

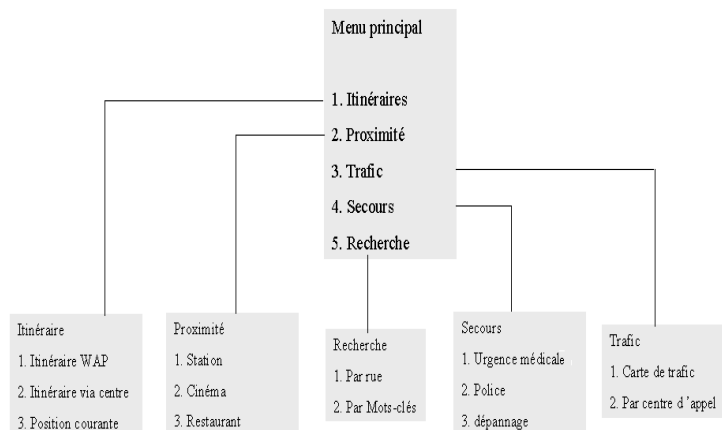


FIG. 2 - *Menu principal*

4 Implantation en LDS de l'application

4.1 Spécification SDL des services de localisation

Comme nous l'avons vu précédemment, notre application repose sur l'utilisation de bases de données. Pour le moment nous n'en utiliserons que trois :

- Base de données qui contient les catégories (exemple cinéma), les types (UGC), les coordonnées (adresse, téléphone) et des informations complémentaires (carte) des lieux.
- Base de données qui contient la position de chaque utilisateur. Comme nous ne pouvons pas accéder au service de localisation (informations véhiculées par le protocole MPP), nous utiliserons une base de données, consulté par le MPC, et où est stockée la position de chaque terminal et qui est mise à jour à chaque instant. Dans notre cas, nous introduirons ces informations nous-mêmes. Le terminal, au moment de choisir le service qu'il désire avoir, communique son numéro SIM² (Subscriber Identification Module) à partir duquel on détermine sa position.
- Base de données qui contient le nom de toutes les rues avec la distance et le chemin entre elles.

Le système a été décrit en utilisant le langage LDS [8], aussi bien pour la partie localisation, que pour l'invocation de services et le comportement du terminal. Vu le nombre de réponses que nous pouvons avoir pour une requête, nous utiliserons une liste. Pour satisfaire ce type de variables, nous avons été obligés d'utiliser les déclarations en ASN.1 [10].

Ce système fonctionne en suivant les échanges d'informations cités auparavant. Il est composé de deux blocs (figure 3):

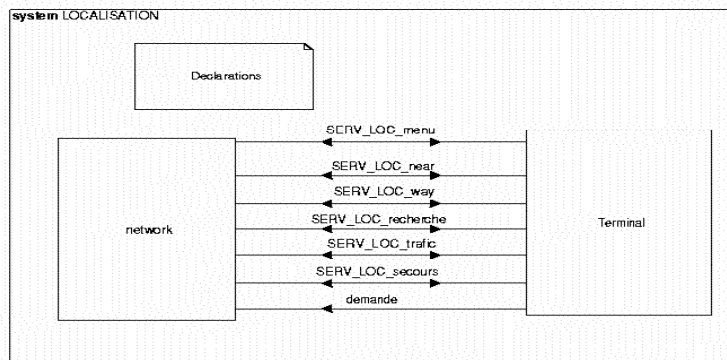


FIG. 3 - *Le système de l'application*

1. Le bloc "Terminal" contient un processus "mobile" qui décrit le comportement du terminal.
2. Le bloc "network" est composé de deux blocs (figure 4) :

2. La carte SIM est une puce qui permet l'utilisation du terminal mobile. Elle contient certain nombre de fonction de sécurité, et d'autres informations comme le numéro de téléphone et le nom de l'utilisateur

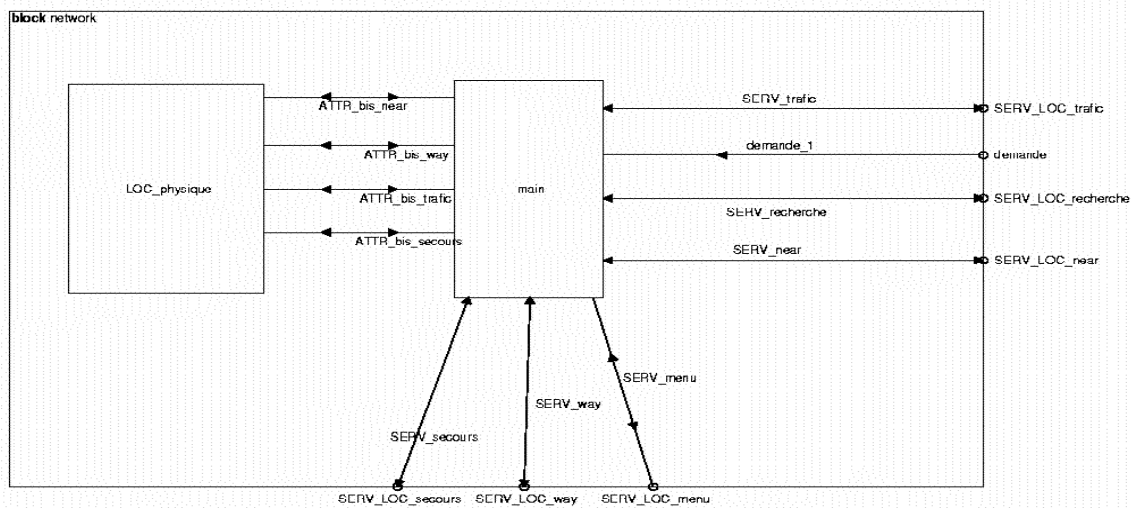


FIG. 4 - Bloc Network

- le bloc “LOC_Physique” composé d’un processus “MPC” qui décrit le comportement du serveur de localisation,
- le bloc “main” qui contient un autre bloc (pour faciliter l’intégration, dans le future, d’un nouveau service) “LOC_Service” qui décrit le comportement du serveur d’application. Ce bloc est composé de six processus, un processus “menu” et les cinq processus des services (Proximité, Itinéraire...)(figure 5).

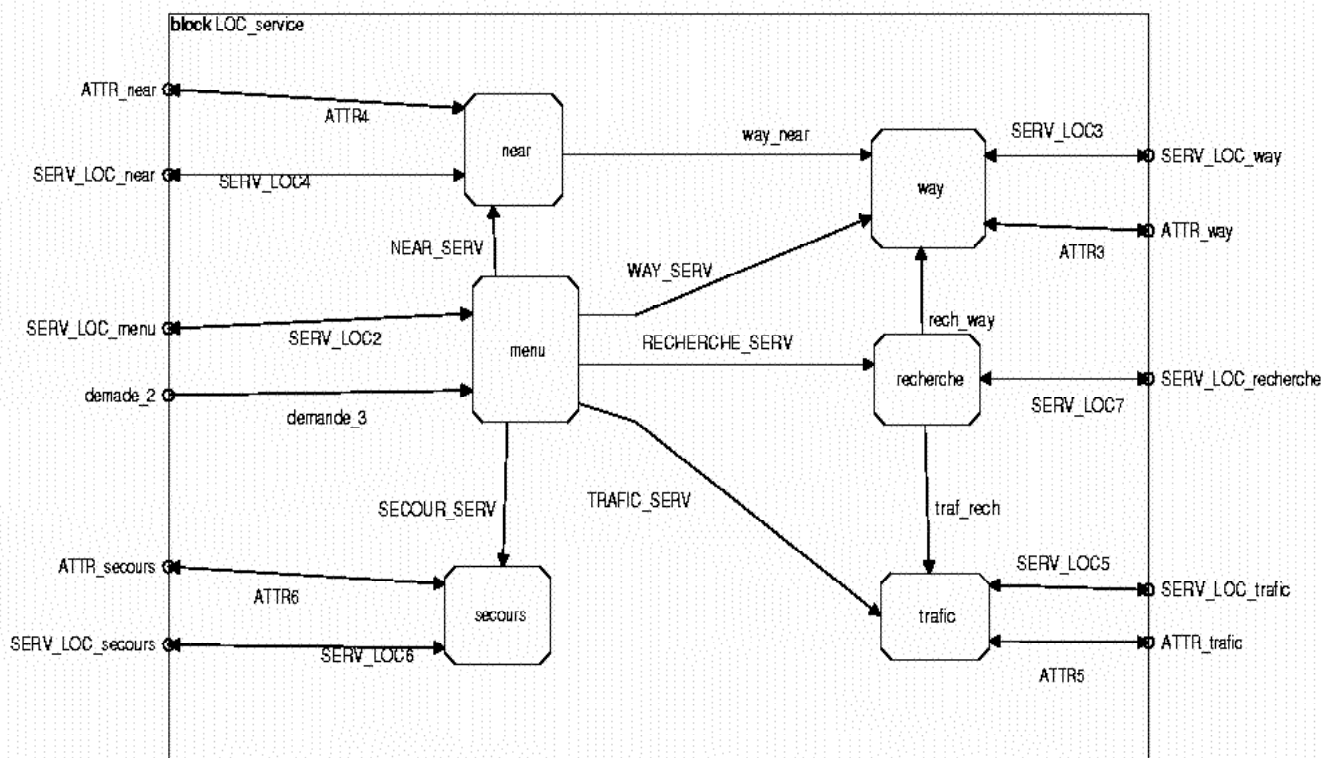


FIG. 5 - Bloc LOC_Service

Le processus menu communique avec les processus des services et le processus mobile (dans le bloc terminal). Le processus MPC communique juste avec les processus des services. Il existe un lien entre les processus service (du bloc LOC_service) et le processus mobile (du bloc terminal).

Les processus des services communiquent entre eux : le processus “near” (service de proximité) sollicite le processus “way” (service d’Itinéraire). Le processus “recherche” sollicite les deux processus “way” et “trafic” (service de trafic). Le processus “secours” (du service de secours) ne sollicite aucun autre processus service.

Plusieurs Fonctions ont été définies dans la spécification pour effectuer certaines tâches redondantes ou pour parcourir les tableaux (bases de données).

Afin de fournir une idée générale de la complexité de la spécification du système LDS, on présente dans la figure 6 quelques chiffres significatifs sur le système global. Ce dernier a été simulé en utilisant le mode de simulation exhaustive, afin d’obtenir un graphe d’accessibilité complet. Afin d’exécuter la simulation du système, on le configure avec un fichier de démarrage, qui initialise certaines variables, telles que les bases de données, le numéro de la carte SIM qui invoque le service, le nom de la rue ou le mot-clé à chercher. La figure 7 donne quelques informations sur le nombre d’états, de transition, etc., tous obtenus après un arrêt manuel de recherche exhaustive. Il est impossible de construire le graphe d’accessibilité complet, ceci étant dû aux énormes ressources mémoires requises.

Lignes	7947
Blocs	5
Processus	8
Procédure	18
Etats	52
Signaux	29
Macro définition	0
Timers	1

FIG. 6 - *Mesures de la spécification.*

Nombre d’états	1109871
Nombre de transitions	3535611
Profondeur maximale atteinte	54
Durée	56 mn 8 s
Couverture des transitions	96.95%
Couverture d’états	97.53%

FIG. 7 - *Simulation partielle de la spécification complète.*

4.2 Génération de test pour le module *Way*

Dans cette partie, nous détaillerons les résultats obtenus par la génération de tests pour le module way, et plus précisément l’itinéraire (rue/rue). Pour le tester, nous utiliserons un algorithme pour le test de composants développé à l’INT [5].

Pour tester le module way, les transitions, décrites en *annexe A* qui fournit la spécification en SDL du module, sont traversées au moins une fois. Les conditions d’arrêt sont représentées dans la figure 8. Les résultats de l’utilisation de l’algorithme sont détaillés dans la figure 9.

Une fois que toutes les transitions du composant imbriqué way sont traversées, on obtient une séquence de test unique, qui correspond à la longueur totale du chemin qui a été traversé de l’environnement à la dernière transition du module. La séquence obtenue a une longueur de 64 transitions.

stop if	input OK_LOC_way to way and output REQ_ATTR_way from way	1
or	input REP_ATTR_way to way and output DATA_LOC_way from way	#2
or	input OK_LOC_way to way and way!requette_way!answe=’it_wap’	#3

FIG. 8 - *Conditions d’arrêt du processus way.*

La figure 10 montre une partie des traces MSC (séquence de test), du service itinéraire, obtenus après application de l’algorithme sous mentionné. Nous constatons que les échanges entre les processus serveur et le processus terminal reflète bien le comportement du service itinéraire précédemment explicité.

Conditions d'arrêt	1	#2	#3
Nombre de transitions	18	21	37
Profondeur atteinte	18	3	16
Durée (secondes)	2	0	1

FIG. 9 - Résultats de la simulation pour chaque condition d'arrêt.

5 Conclusion

Nous avons présenté dans cet article les services basés sur la localisation appliqués dans le domaine de la téléphonie mobile, et qui sont étudiés dans le cadre du projet PLATONIS. Ces services permettent d'acheminer l'information et de la mettre à la disposition de l'utilisateur en se basant sur son emplacement à tout moment. Ces derniers ont été formellement décrits en utilisant une technique de description formelle, à savoir le langage LDS.

Enfin, des tests sur ces services ont été effectués afin de vérifier leurs comportements par rapport à leurs descriptions formelles. La description formelle de ces services a permis de générer automatiquement un ensemble de scénarios, en utilisant un algorithme pour le test de composants.

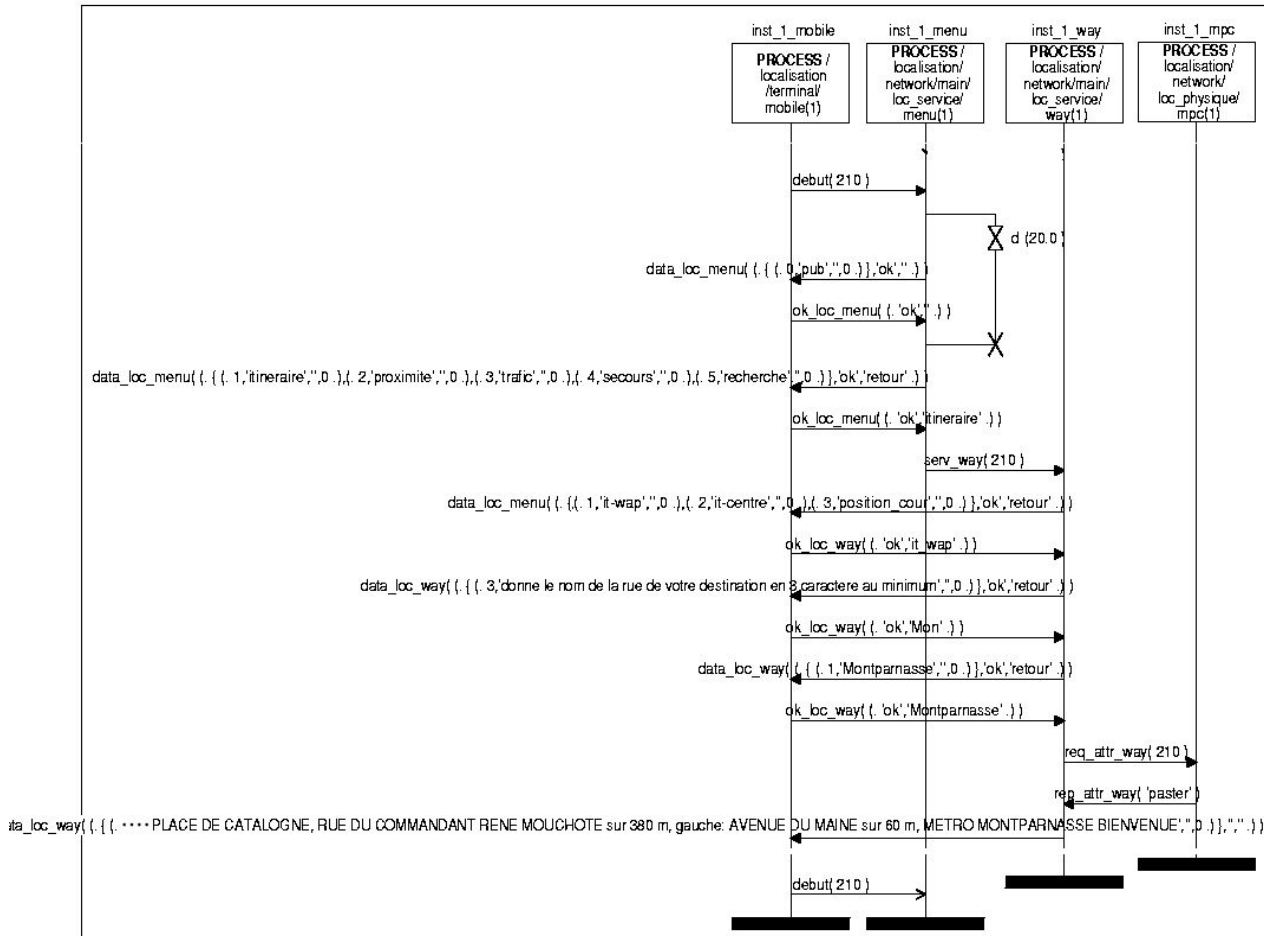


FIG. 10 - MSC du service Itinéraire WAP

Références

- [1] <http://www.wirelessdevnet.com/channels/lbs/features/lbsinterfacing.html>.
- [2] <http://www.wirelessdevnet.com/channels/lbs/features/mobilepositioning.html>.
- [3] <http://www.jlocationsservices.com/GettingStarted/tutorials/WhatAreLocationServices.html>.
- [4] <http://www.mappy.fr>.
- [5] A.Cavalli, D.Lee, C.Rinderknecht, and F.Zaïdi. Hit-or-jump: An algorithm for embedded testing with applications to in service. *FORTE/PSTV'99*.
- [6] The PLATONIS Consortium. The platonis project. In *First International Workshop on Services Applications in the Wireless Public Infrastructur*, Mai 2001. <http://www-lor.int-evry.fr/platonis>.
- [7] Ericsson. Mobile positioning protocol version 1.1. Technical report, 2000.
- [8] ITU. *CCITT Specification and Description Language(SDL)*. Recommendation Z.100, 1992.
- [9] Marcien Mackaya. Le wap architecture et applications. juin 2001.
- [10] O.Dubuisson. *ASN.1*. Springer, 1999.
- [11] Randolph Wohler. Location based services service requirements document. January 2000. http://www.gsmworld.com/technology/mls_presentations.html.

A Processus way

