

UNIVERSITE DE GENEVE
FACULTE DES SCIENCES ECONOMIQUES
ET SOCIALES

Département des Systèmes d'Information et de Communication

DISPATCH-LOCATOR

Joëlle Levi et Stéphane Krattinger

Projet de bachelor
Octobre 2006

Professeur : Dimitri Konstantas

Superviseur : Michel Deriaz



Table des matières

CHAPITRE 1 : INTRODUCTION	3
CHAPITRE 2 : MARKET SURVEY	6
CHAPITRE 3 : TECHNOLOGIES	10
3.1 JAVA	10
3.2 GPS	15
3.3 BLUETOOTH	18
3.4 GPRS	19
3.5 AUTRES NORMES DE TRANSMISSION	20
3.6 ADSL ET WIFI	21
CHAPITRE 4 : ARCHITECTURE	23
4.1 SCHEMA GENERAL	23
4.2 COMMUNICATIONS	24
4.3 LE SERVEUR	25
4.4 L'APPLICATION COORDINATEUR	25
4.5 L'APPLICATION MOBILE	26
CHAPITRE 5 : MANUELS D'UTILISATION	27
5.1 L'INTERFACE COORDINATEUR	27
5.2 LE MOBILE	38
CHAPITRE 6 : CAS D'UTILISATION – COMMUNICATION TRIPARTITE	47
CHAPITRE 7 : COLLABORATION AVEC LES SAMARITAINS	50
7.1 OBSERVATION : LAKE PARADE	50
7.2 DEPLOIEMENT : FETES DE GENEVE	52
CHAPITRE 8 : EVOLUTIONS FUTURES	55
CHAPITRE 9 : CONCLUSION	58
REMERCIEMENTS	60
REFERENCES	61

Chapitre 1 : Introduction

Depuis plusieurs années, les systèmes de géolocalisation connaissent un essor fulgurant auprès du grand public. Aujourd'hui, du fait de l'expansion des réseaux sans-fil et de l'amélioration technique des appareils mobiles tels que les téléphones ou les PDA qui possèdent une plus grande puissance de calcul, une capacité de stockage croissante (mémoires auxiliaires), une meilleure connectivité et l'intégration de Java, il est possible de coupler un simple récepteur GPS à un appareil mobile. Ce procédé permet de bénéficier d'informations telles que la position, qui sert par exemple à se repérer sur une carte ou à suivre un itinéraire.

Assistant à l'Université de Genève au Département des Systèmes d'information et communication (groupe ASG), Michel Deriaz, fort de son expérience de samaritains lors de grands événements, a émis l'idée de combler les lacunes du système de gestion des équipes de samaritains sur le terrain, grâce à leur repérage en temps réel sur une carte géographique via un téléphone mobile et un récepteur GPS.

A partir de cette idée, nous avons décidé de développer pour notre travail de Bachelor, un système de gestion et de déploiement du personnel par géolocalisation. Nous nous sommes fondés sur les besoins des samaritains dont nous avons été observer le travail lors de la Lake Parade 2006 à Genève. Cette observation nous a servi de point de départ pour comprendre les fondements d'une telle gestion. Notre système a été conçu dans l'optique de pouvoir être utilisé dans des situations très diverses. De plus, afin d'être facilement accessible et abordable par tous, nous avons voulu notre système open-source et gratuit.

Dispatch-Locator est composé de trois parties : une application coordinateur, une application mobile pour les équipes sur le terrain et un serveur. L'idée de base a été sensiblement étoffée, pour permettre entre autre l'envoi de missions de demande d'intervention ou de remplacement sur le terrain et la connaissance du statut de

chaque équipe. Notre architecture permet en outre d'archiver l'ensemble des informations (identités et missions) de la manifestation afin de réduire la charge qu'impose la maintenance d'une main courante.

Pour plusieurs raisons, nous avons opté pour l'utilisation du téléphone mobile pour les équipes sur le terrain. Tout d'abord, la majorité des personnes en possèdent un et savent l'utiliser correctement, ce qui pose moins de blocages dus à la nouveauté. De plus, les opérateurs téléphoniques ont largement étendu la couverture du réseau GPRS, qui est un excellent moyen et à moindre frais pour communiquer avec un serveur distant via Internet. En outre, il existe une version réduite de Java (J2ME) destinée aux appareils mobiles et présente sur de nombreux modèles, ce qui nous a permis d'utiliser ce langage pour développer notre application. Finalement, Stéphane Velen et Christophe Praplan, deux anciens étudiants à l'Université de Genève, ont créé une API (Application Programming Interface)* pour téléphones mobiles en Java qui permet d'établir une connexion Bluetooth avec un récepteur GPS et de traiter les données provenant de celui-ci.

Le langage Java s'est donc imposé pour ces raisons et parce qu'il est extrêmement répandu. Ceci est également un avantage pour l'application destinée au coordinateur qui peut dès lors, fonctionner sur n'importe quel ordinateur en minimisant les problèmes de compatibilité.

De plus en plus d'applications basées sur ce système voient le jour. A notre connaissance, il n'existe qu'une seule application sur le marché qui propose certaines fonctionnalités similaires aux nôtres. Nous allons donc commencer par décrire ce produit et montrer en quoi il diffère du notre. Puis nous expliquerons les technologies que nous avons utilisées pour réaliser ce projet et présenterons l'architecture de Dispatch-Locator. Ensuite, nous détaillerons les modes d'emplois de l'application coordinateur et de celle pour les téléphones mobiles. Des schémas suivront, qui décrivent plus spécifiquement les liens ainsi que les relations de

* www.universal-locator.com

causes à effets entre les événements générés par les deux applications. Enfin, nous parlerons de notre collaboration avec les samaritains de Genève lors de deux manifestations estivales et finalement des évolutions qui seraient envisageables pour améliorer notre système.

Chapitre 2 : Market Survey

GeoGeny

Geogeny* est le produit phare de la société CPR Groupe (Suisse) basée à Montreux. Le but de ce produit est de permettre d'établir un lien entre un centre de contrôle et différents intervenants d'une équipe sur le terrain. Pour ce faire, ils utilisent la localisation par GPS avec la norme "SIRF Star III", qui leur offre une assez bonne précision. Ainsi, ils peuvent suivre les unités itinérantes qui sont sur le terrain et communiquer avec elles via l'échange de données et de voix (fig.1). Les données (de coordonnées) sont cryptées et transmises par SMS (Short Message Service) via GSM (Global System for Mobile Communication). Pour l'instant, les normes GPRS (General Packet Radio Service) – qui permet d'augmenter la vitesse de transmission des données sur le réseau GSM – et UMTS (Universal Mobile Telecommunications System / 3G) sont en phase de développement.



Fig.1 Mallette de commandement GeoGeny

* www.geogeny.ch

Leur système peut fonctionner sur plusieurs types de terminaux mobiles (fig.2). En partenariat avec HP et Secufone AG, ils utilisent des terminaux professionnels spécifiques incluant leur logiciel, dont ils disposent d'une réserve de 10'000 pièces. Le Secufone est un téléphone mobile GSM tribande, ne supportant pas l'UMTS et doté d'un récepteur GPS intégré. En cas d'urgence, un bouton rouge est présent sur le dessus de l'appareil qui, une fois pressée, transmet une alarme à la centrale (fig.3).



Fig.2 PDA et Secufone avec l'application GeoGeny



Fig.3 Le bouton rouge d'alarme du Secufone

GeoGeny est un système propriétaire et payant (forfait initial de 90 frs. par terminal + prix des communications). La mise en place et la configuration doivent

obligatoirement être faites par leurs soins. De plus, comme les terminaux sont loués, il est impératif de les réserver à l'avance ce qui peut poser des problèmes de flexibilité. A titre d'exemple, le Montreux Jazz Festival a dû réserver plus de 2 mois à l'avance le système GeoGeny.

Le système que nous proposons est libre et donc gratuit. En outre, Dispatch-Locator a été réalisé en Java, qui a la particularité d'être portable sur plusieurs systèmes d'exploitation tel que Windows, MacOS ou Linux (cf. 3.1). Il ne nécessite qu'un serveur Apache Tomcat (libre, gratuit et multi-plateforme) qui peut tourner sur n'importe quel ordinateur relié à un réseau (Internet ou local). La facilité d'installation et de configuration permet de l'installer très rapidement et sans assistance particulière afin d'obtenir un serveur indépendant.

Pour utiliser notre système, il faut tout d'abord télécharger l'application pour le coordinateur ainsi que celle pour les terminaux mobiles sur notre site web (www.dispatch-locator.com). Il n'y a pas besoin d'installer ni de configurer l'application coordinateur, il suffit de la lancer sur n'importe quel ordinateur. Dans le cas où celui-ci ne peut accéder à Internet (par exemple s'il n'a pas d'accès Ethernet ou Wi-Fi), il est possible de se connecter au moyen d'une carte GPRS ou de se servir d'un téléphone portable comme d'un modem (moyennant une liaison USB ou Bluetooth). Ainsi, en cas de problème technique, il n'est pas nécessaire d'avoir à attendre plusieurs heures qu'un technicien vienne depuis Montreux (siège de la société CPR Group) réparer le système. Il suffit de trouver un autre ordinateur puis de relancer l'application coordinateur. Les paramètres étant conservés sur le serveur rien n'aura été perdu.

Les terminaux du personnel itinérant sont, pour l'instant, des téléphones mobiles. La seule nécessité est qu'ils possèdent la technologie Java ainsi que les connectivités Bluetooth et GPRS, ce qui est le cas sur la plupart des appareils récents. Cette flexibilité permet d'utiliser Dispatch-Locator sur les téléphones déjà possédés par les membres de l'équipe sans avoir à investir ni à louer de nouveaux terminaux comme ce serait le cas avec GeoGeny. L'application étant téléchargeable en tout temps sur Internet directement depuis un téléphone mobile,

il est facile de répondre à un besoin croissant et immédiat de terminaux mobiles supplémentaires. Le coordinateur n'a alors plus qu'à inscrire les nouvelles identités dans le fichier de données via l'interface coordinateur de Dispatch-Locator, pour que les utilisateurs aient instantanément accès à l'ensemble du programme.

La connexion entre le téléphone mobile et le serveur est permanente grâce à une liaison GPRS, ce que GeoGeny ne propose pas. Notre système bénéficie d'une grande transparence quant à la technologie utilisée car les téléphones mobiles peuvent être connectés indifféremment par GPRS ou par UMTS lors d'une même manifestation. Il est à noter qu'au jour d'aujourd'hui, nous préférons le GPRS à l'UMTS principalement à cause de la couverture réseau. Ce choix est laissé à la discrétion de l'utilisateur en fonction de ses besoins et de ses moyens financiers. La seule nécessité est d'être connecté à l'Internet.

Enfin, la licence LGPL que nous avons mis sur l'ensemble du code source de Dispatch-Locator offre à tout le monde, comme on l'a vu, la possibilité d'utiliser notre application mais également de la modifier à sa guise et de la sécuriser selon ses propres critères.

Chapitre 3 : Technologies

3.1 Java

3.1.1 J2SE (Java 2 Standard Edition)

Java est à la fois un langage de programmation (orienté objet) et un environnement d'exécution (JRE – Java Runtime Environnement) basé sur une machine virtuelle (JVM – Java Virtual Machine) qui permet d'interpréter et d'exécuter le bytecode Java.

Le principal avantage de Java est sa portabilité. En effet, il suffit de disposer de la JVM correspondant au système d'exploitation utilisé (Windows, MAC OS, Linux) pour pouvoir correctement exécuter une application développée en Java. Afin de pouvoir exécuter l'application coordinateur de Dispatch-Locator, il faut que le JRE soit installé sur la machine. Celui-ci est téléchargeable sur le site de Sun^{*}. Pour des raisons de comptabilité, il est nécessaire d'avoir la dernière version du JRE (5.0).

3.1.2 J2ME (Java 2 Mobile Edition)

J2ME est la version de Java pour les terminaux mobiles (téléphones mobiles, PDA...) et les systèmes embarqués (carte à puce, décodeur TV numérique...).

Pour fonctionner, trois composants sont nécessaires :

- une machine virtuelle pour exécuter l'application appelée KVM (Kilobyte Virtual Machine), qui est une version très réduite de la machine virtuelle Java JVM suffisamment légère pour fonctionner sur des terminaux disposant de peu de mémoire.
- une configuration qui est une librairie (API) permettant d'accéder aux fonctions de base du système. Actuellement il y en a deux : le Connected

^{*} <http://java.sun.com>

Device Configuration (CDC) pour les appareils de plus grande capacité et le Connected Limited Device Configuration (CLDC), qui comme son nom l'indique, est utilisé sur les plus petits appareils aux capacités limitées.

- un profil qui est une spécification de l'API Java qui diffère selon chaque constructeur. Il en existe plusieurs types, notamment celui pour les téléphones mobiles qui est le Mobile Information Device Profile (MIDP). Les applications fondées sur ce profil s'appellent des MIDlets. Le fait qu'il existe presque autant de versions différentes de MIDP que de marques de téléphones mobiles, pose des problèmes de gestion des threads et de gestion de l'affichage. Ainsi, une MIDlet peut avoir des comportements très différents selon le téléphone sur lequel elle est exécutée (Sony-Ericsson, Nokia, etc.).

Pour pouvoir correctement exécuter notre application sur un téléphone mobile, il faut que celui-ci ait le CLDC 1.1 et le MIDP 2.0. Nous avons principalement testé notre application sur le Sony-Ericsson K750i, mais il existe un grand nombre de téléphones mobiles qui devraient être compatibles.

3.1.3 Définition d'un thread

Un thread (ou "processus léger") est en quelque sorte un processus à l'intérieur d'un autre. Chaque processus possède au moins un thread qui exécute le programme principal (fonction *main()* en Java). S'il y a plusieurs threads, les ressources qui sont allouées à un processus (temps et mémoire) sont partagées entre ces différents threads qui le composent.

3.1.4 Les servlets HTTP

A. Protocole HTTP (HyperText Transfer Protocol)

Ce protocole est utilisé partout sur Internet, c'est lui qui permet à tout un chacun de voir des pages web. Il fonctionne de manière très simple : lorsque

l'utilisateur entre une URL (Uniform Resource Locator) dans la barre d'adresse, une requête est envoyée au serveur. Ce dernier répond en renvoyant les informations demandées (en règle général sous forme de code HTML qui permet d'afficher une page web). Une requête est composée de trois parties : la commande (commande HTTP, URL et version du protocole ex: *GET* /www.unige.ch http v1.01), l'entête qui contient des renseignements sur l'émetteur, le style de texte et différents autres paramètres et enfin le corps de la requête qui contient les données. Il existe deux formes de requêtes ou méthodes qui ne sont pas forcément utilisées de la même manière (en fait, il y en a plus mais les autres sont trop peu utilisées pour que nous les traitions ici). Premièrement, la méthode *get* qui est une simple URL à laquelle on ajoute une chaîne d'interrogation (précédée d'un point d'interrogation, ex : http://www.mondomaine.com?arg1=data1&arg2=data2). La chaîne d'interrogation permet de préciser la requête en lui rajoutant des paramètres que le serveur peut récupérer et qui, en fonction, répond de manière différente. L'inconvénient de cette méthode est que tous les arguments sont contenus dans l'URL et sont donc facilement récupérables par l'utilisateur. La méthode *get* peut présenter un problème si l'on veut envoyer beaucoup d'informations car il faut construire une URL extrêmement longue et la plupart des serveurs limitent cette longueur à 240 caractères.

Il existe également une deuxième méthode : la méthode *post*, qui permet d'envoyer autant de données que l'on veut car elle sont envoyées directement sur la socket dans le corps de la requête HTTP. L'échange est donc invisible pour l'utilisateur car, contrairement à la méthode *get*, elle ne réécrit pas l'URL. Cela évite aussi que des informations puissent être envoyées deux fois car l'utilisateur n'a nulle part la possibilité de récupérer les données transmises avec la méthode *post* dans le logiciel pour le coordinateur.

B. XML (eXtensible Markup Language)

Ce langage a été élaboré par le World Wide Web Consortium (W3C^{*}) pour pouvoir décrire les données mises en page dans les autres langages balisés comme le HTML (Hyper Text Markup Language). XML a la particularité, contrairement au HTML, de ne pas être limité par un certain nombre de balises pré-existantes, puisque celles-ci sont définies par l'utilisateur lui-même. Ce langage est composé de balises qui décrivent la donnée contenue entre elles, comme par exemple :

```
<Personne>  
  <Nom> Ceci est le nom de la personne </Nom>  
</Personne>
```

L'exemple indique que ce que l'on va trouver entre les balises "Nom" est le nom de l'objet courant (ici une personne). Il est possible de définir un schéma hiérarchique qui sera appliqué à chaque nouvel élément inscrit dans le fichier. Ainsi, tous les objets contiendront le même ensemble de balises. Pour être sûr que le fichier soit bien formaté, c'est-à-dire qu'il comporte les bonnes balises dans le bon ordre, il est possible d'écrire un autre document appelé DTD (Document Type Definition) qui indique comment et quelles balises doivent être utilisées. Cela limite artificiellement le nombre de balises qui pourront être employées et permet de pouvoir transmettre les documents entre les applications plus facilement.

Pour l'instant, la DTD des fichiers nécessaires à notre projet n'existe pas encore mais la structure y est assez simple. Une évolution future sera d'en créer une afin de garantir un formatage correct.

* www.w3c.org

C. Les servlets

Sun Microsystems à muni le langage Java de ce qui s'appelle le "côté serveur", c'est-à-dire des classes qui sont capables de se comporter comme un serveur. Ces classes sont appelées "servlet". Dans le cadre de ce projet nous utilisons des servlets HTTP c'est-à-dire des servlets permettant de traiter les requêtes HTTP comme le ferait un serveur web normal (cf. 3.1.4.A). La puissance des servlets réside dans le fait que pour une seule classe se crée une multitude de threads (cf. 3.1.3) qui sont indépendants et gèrent donc séparément les requêtes qui arrivent au serveur. Il n'y a donc pas d'attente car pour chaque nouvelle connexion au serveur, celui-ci crée un thread autonome issu de la classe originale (bien sûr dans la limite des capacités du serveur). Dès lors, il est impératif de gérer correctement les objets communs pour toutes les requêtes en synchronisant leurs accès. Dans notre projet, les fichiers XML servant de base de données sont accédés tant par le coordinateur que par les utilisateurs mobiles ; il est donc impératif que lorsque l'un veut lire un des fichiers, l'autre ne soit pas en train d'écrire dessus, sinon les données lues par le premier ne seront pas correctes et vice versa. La puissance des servlets réside dans les méthodes prédéfinies du langage Java qui permettent de traiter au mieux et au plus vite les requêtes. Il existe par exemple une méthode qui acquière automatiquement les paramètres de n'importe quelle requête *get* ou *post*.

D. Serveur Tomcat

Toutes les servlets ont besoin pour fonctionner d'un conteneur qui les lance et qui gère les multiples threads qui lui sont attachés. Dans notre cas, nous avons choisi le conteneur d'Apache : Tomcat^{*}. Il est très simple à installer et à gérer. Il existe d'autres conteneurs y compris un fourni par Sun dans la version Entreprise de Java (J2EE^{**}). L'Université de Genève nous a permis

* <http://tomcat.apache.org/> ** <http://java.sun.com/javaee/>

d'héberger notre serveur dans le parc informatique de l'Université pour réaliser les tests et les premières mises en service, ce qui nous garantit une permanence du service et une maintenance sûre.

3.2 GPS

Un des buts de notre système est de pouvoir localiser les membres d'une équipe qui sont sur le terrain. Pour ce faire, nous utilisons le GPS (Global Positioning System) qui est un système développé par le Département de la Défense des Etats-Unis afin de pouvoir connaître la position d'un objet sur la surface de la Terre avec une précision d'environ 10 mètres. Le système GPS fonctionne grâce à 24 satellites placés en orbite autour de la Terre et dont la position est connue à tout moment (fig. 4).



Fig.4 Satellites GPS autour de la Terre

La surface de la Terre est découpée par des méridiens – lignes virtuelles qui lient les deux pôles - et des parallèles qui leurs sont perpendiculaires (fig.5). Tous les points qui sont situés sur un même méridien ont la même longitude et le méridien de référence est celui de Greenwich (méridien zéro), qui passe par l'observatoire de Greenwich, en Angleterre. De même, tous les points situés sur un même parallèle ont la même latitude. Le parallèle de référence est le plus grand : l'Equateur.

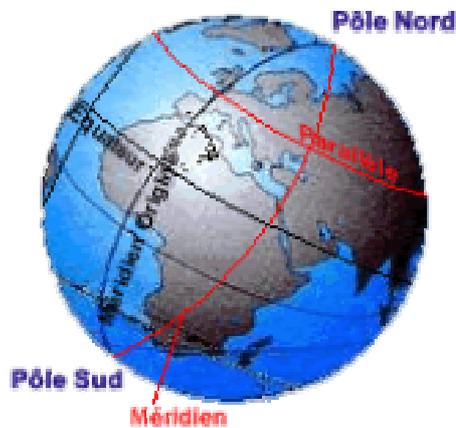


Fig.5 Parallèle et Méridien

Chaque point sur la Terre se caractérise par des coordonnées géographiques latitude – longitude. La latitude est l'angle allant de 0° à l'équateur à 90° aux pôles (fig.6a). La longitude est l'angle allant du méridien de référence à 180° (fig.6b).

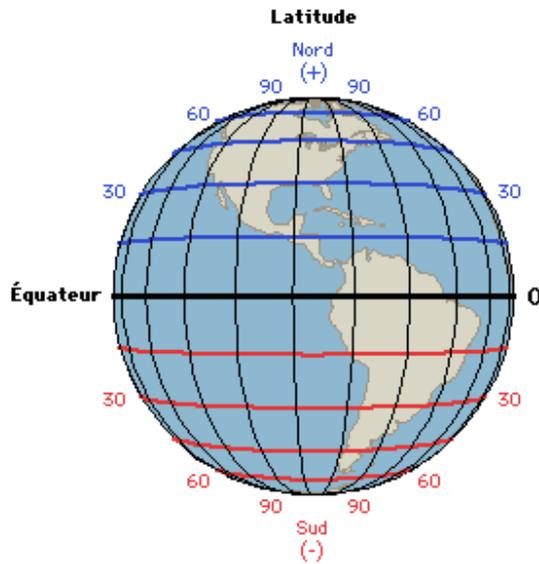


Fig.6a Latitude

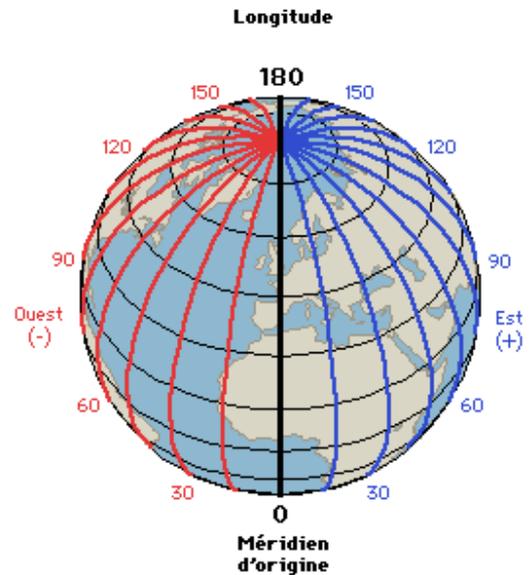


Fig.6b Longitude

Pour fournir une position (ou FIX) latitude - longitude, le récepteur doit capter les signaux d'au moins 3 satellites afin de déterminer l'intersection des sphères centrées sur ces satellites (fig.7). Les coordonnées sont exprimées selon le système géodésique WGS84 (World Geodetic System, révision de 1984), qui est le système de référence pour exprimer les positions en degrés décimales au voisinage de la Terre et par conséquent celui du GPS.

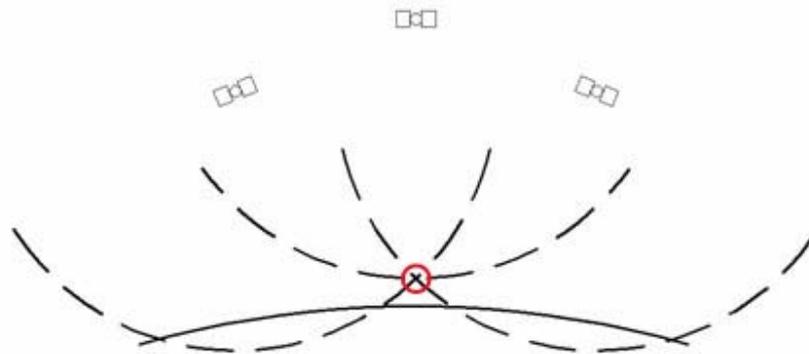


Fig.7 L'intersection donne la position

Les informations minimales nécessaires au bon fonctionnement de notre application fournies par le GPS sont la latitude, la longitude et l'azimut qui est l'angle formé entre le Nord géographique et la direction de l'utilisateur.

Aujourd'hui, de plus en plus de récepteurs GPS fonctionnent avec une puce de type SIRF Star III, qui permet de capter un signal GPS dans des conditions dites "extrêmes" (intérieurs, derrière un pare-brise athermique,...).

3.3 Bluetooth



Le Bluetooth est une technologie à ondes radio courte distance qui permet d'établir une connexion entre des appareils électroniques sans utiliser de câble ni d'autre infrastructure. La portée de ces ondes est d'environ 10 mètres (non sécurisé).

Cette liaison Bluetooth permet au terminal mobile de communiquer avec le récepteur GPS (fig.8). Cette technologie est aussi utilisée pour connecter les oreillettes sans fil des téléphones portables.



Fig.8 Connexion Bluetooth

3.4 GPRS

Le réseau GSM est une norme numérique de 2^{ème} génération avant tout destinée aux communications vocales. Lors d'une transmission, les ressources du réseau ne sont allouées que pour la durée de la conversation (réseau commuté) comme pour le téléphone fixe.

Le GPRS (General Packet Radio Service) est une norme de génération 2,5 pour la téléphonie mobile basée sur le réseau GSM (Global System for Mobile Communications). La transmission de données se fait cette fois par paquets (comme sur Internet) et les ressources du réseau ne sont mobilisées que lors d'une transmission effective de données. L'utilisateur reste atteignable en permanence grâce à une adresse IP (Internet Protocol) mais ne paye pas pour le temps de connexion mais pour le volume de données transmises. Un autre avantage du GPRS est la possibilité d'utiliser plusieurs canaux GSM à la fois afin d'augmenter la vitesse de transmission. En Suisse, les principaux opérateurs proposent automatiquement le GPRS à leurs clients, sans frais de base supplémentaires.

De nombreux terminaux mobiles actuels supportent cette connectivité rapide, efficace et à moindre frais. Il est également possible d'employer le GPRS depuis un ordinateur portable en utilisant soit une carte externe (généralement PCMCIA), soit un téléphone mobile comme modem. Ce sont pour ces raisons que nous privilégions, aujourd'hui, ce moyen de transmission.

Tarif des 3 principaux opérateurs en Suisse :

Swisscom : CHF 0.10 / 10 Ko (unité de facturation minimale par session: 10 Ko)

Orange : CHF 0.15 / 10 Ko (unité de facturation minimale par session: 1 Ko)

Sunrise : CHF 0.05 / 10 Ko (unité de facturation minimale par session: 10 Ko)

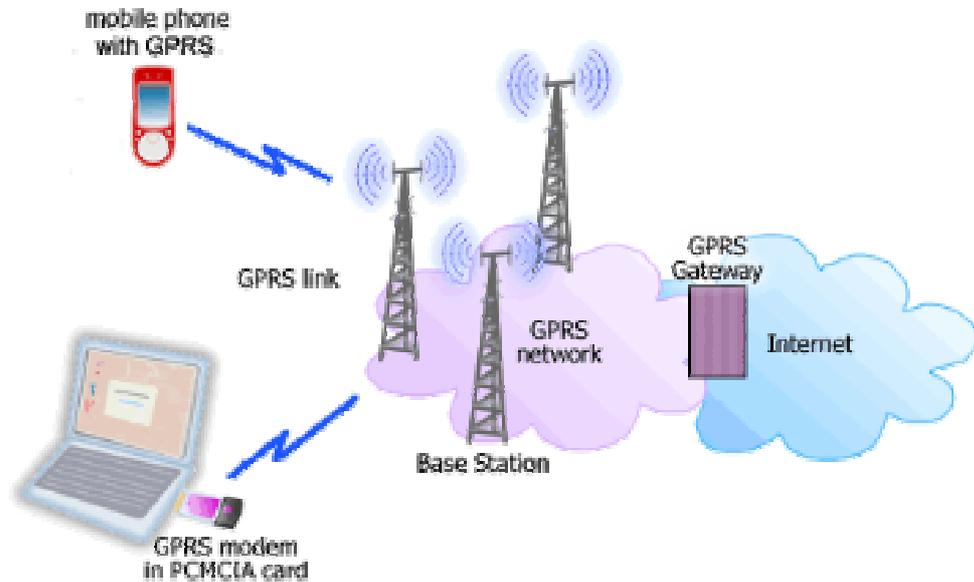


Fig.9 Schéma du fonctionnement général du réseau GPRS

3.5 Autres normes de transmission

Il existe plusieurs autres normes de transmission qui permettent d'obtenir des débits plus élevés parmi lesquels l'UMTS (Universal Mobile Telecommunications System).

Il s'agit d'une norme de 3^{ème} génération, actuellement en plein essor, qui permet d'exploiter une bande de fréquence plus large afin de faire transiter d'avantage de données et d'obtenir par là un débit nettement plus important.

Mais cette technologie a plusieurs désavantages :

- la nécessité d'avoir un terminal mobile très récents et de 3^{ème} génération
- la couverture réseau qui n'est que sommaire

Tableau comparatif des technologies réseau de téléphonie mobile

Technologie	Génération	Débits théoriques maximum
GSM (Global System for Mobile Communications)	2	9,6 Kbps
GPRS (General Packet Radio Service)	2,5	115 Kbps
EDGE (Enhanced Data for GSM Evolution)	2,75	384 Kbps
UMTS (Universal Mobile Telecommunications System)	3	2 Mbps

3.6 ADSL et WiFi

ADSL signifie Asymmetric Digital Subscriber Line, en français : raccordement numérique asymétrique (RNA). C'est un moyen de transmettre des données sur un raccord téléphonique conventionnel. Le terme asymétrique (Asymmetric) désigne le fait que l'on n'a pas la même bande passante montante et descendante (upstream et downstream). En Suisse, les fournisseurs ADSL proposent entre 2 et 5 Mbit/s en descente (downlaod) et 100 et 300 kbit/s en montée (upload) alors que techniquement, l'ADSL permet d'obtenir des débits jusqu'à 8 Mbit/s en descente selon l'éloignement du central téléphonique (la perte de vitesse est due à l'atténuation du signal électrique sur la ligne porteuse, conséquence de l'utilisation des hautes fréquences). Il faut néanmoins munir la prise murale d'un filtre afin de bien séparer les deux franges de fréquences (en vert et en rouge sur la figure 10).

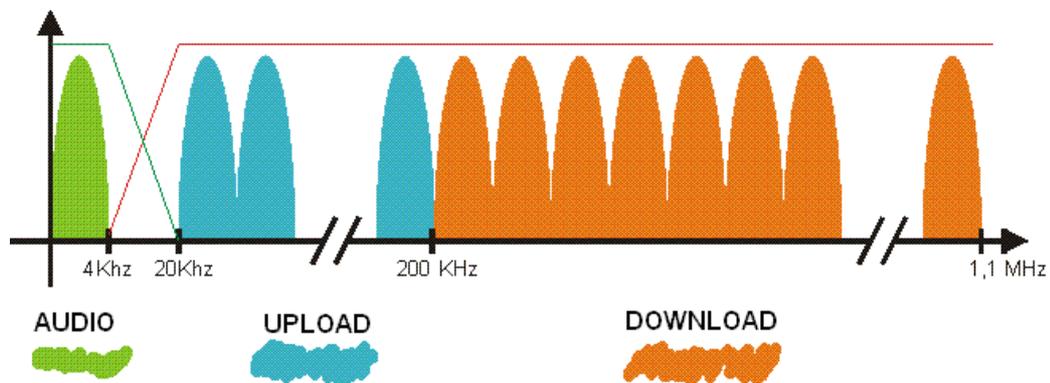


Fig. 10 Schéma de répartition des fréquences pour l'ADSL.

Il existe l'ADSL2 et l'ADSL2+ qui occupent plus de fréquences, et surtout celles très élevées (1 à 2 MHz), alors que l'ADSL utilise les fréquences plus basses et proches de celle du téléphone (jusqu'à 1104 kHz). Ces deux technologies sont destinées à augmenter le débit de la ligne mais sont limitées par le bruit et l'atténuation due à l'utilisation des hautes fréquences plus sensibles à ce genre de phénomènes. Ainsi si l'utilisateur se trouve à moins de 3000 mètres du central, il pourra bénéficier d'un débit pouvant aller jusqu'à 25 Mbit/s en descente et jusqu'à 1 Mbit/s en montée. Si ce n'est pas le cas, l'utilisateur ne recevra plus que l'équivalent de l'ADSL simple.

Le WiFi signifiant *Wireless Fidelity*, est le nom donné par la WECA (Wireless Ethernet Compatibility Alliance qui gère l'interopérabilité entre les matériels répondant à la norme 802.11) à la norme ISO 802.11 qui régit les WLAN (Wireless Local Area Network) ou en français : réseau local sans fil. Cette norme a plusieurs sous normes : la 802.11b qui augmente la vitesse jusqu'à 11 Mb/s, la 802.11g qui arrive à la vitesse de 54 Mbit/s qui est la norme actuelle qui équipe tous les matériels récents. Une norme *n* est attendue pour 2007 et devrait permettre des débits jusqu'à 540 Mbit/s. Contrairement au Bluetooth, le WiFi permet d'obtenir des communications sécurisées.

Chapitre 4 : Architecture

4.1 Schéma général

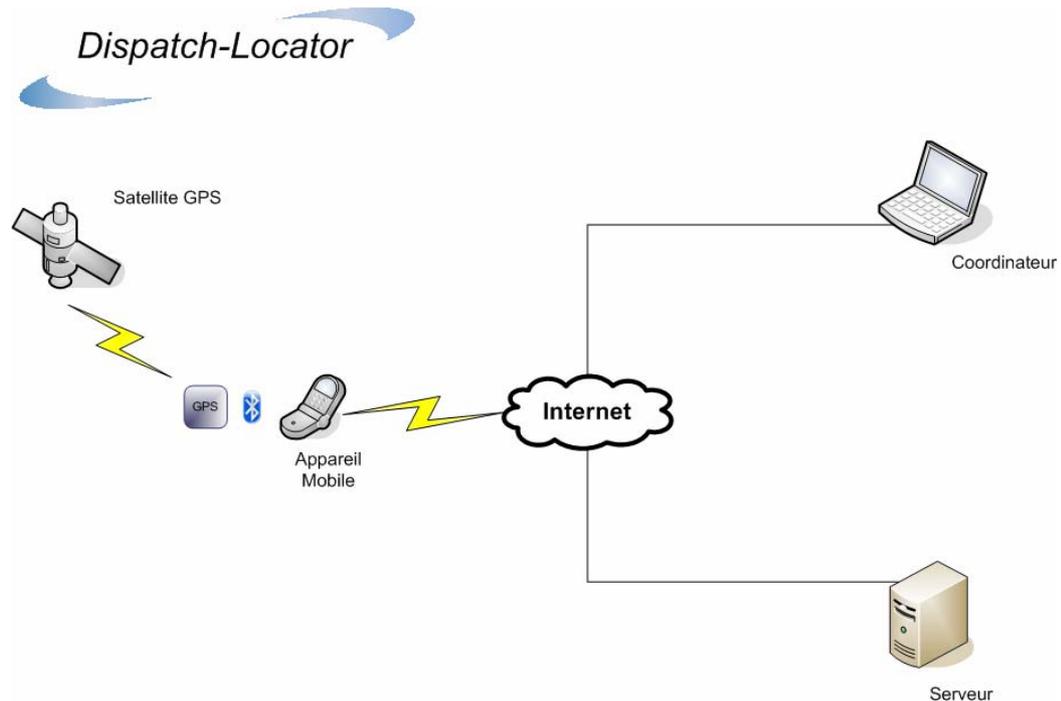


Fig. 11 Architecture générale de Dispatch-Locator

Nous avons conçu une architecture en trois pôles (fig.11) : un pôle mobile constitué d'un appareil portable et d'un récepteur GPS, un pôle coordinateur (fixe) qui se matérialise sur notre schéma par un ordinateur portable (mais qui peut bien sûr être un ordinateur fixe selon les besoins), et enfin un pôle serveur qui joue le rôle de relais entre les appareils mobiles et le coordinateur fixe. Cette architecture a été privilégiée car elle permet de continuer à traiter les événements mobiles même si le coordinateur n'est pas en ligne car le serveur reste actif et inversement. On aurait pu mettre le serveur directement dans le programme coordinateur mais à ce moment il n'aurait plus été possible d'avoir plusieurs coordinateurs ou plusieurs "viewer" (permettant de voir la même chose que le coordinateur mais sans droits

d'interaction) sur la même manifestation (cf. chap. 8 : "Evolutions futures"). De plus, cette architecture permettra, à terme, de faire tourner sur un même serveur plusieurs équipes de différentes manifestations de manière tout à fait indépendante. Cette architecture permet aussi d'éviter d'avoir à installer et à configurer un serveur et de ne pas trop surcharger le portable (ou le fixe) du coordinateur qui pourrait ne pas avoir les capacités matérielles suffisantes pour héberger les deux services (serveur et application coordinateur).

4.2 Communications

GPS → Mobile

L'appareil mobile reçoit des trames GPS (cf. 3.2) depuis un récepteur avec lequel il communique via une connexion Bluetooth (cf. 3.3). Afin de garantir une fiabilité et une précision suffisante, il faut que les données proviennent d'au moins 3 satellites, sans quoi il n'y a pas de FIX. (cf. 3.2)

Mobile → Serveur

Le téléphone portable communique avec le serveur en utilisant le réseau GPRS (cf. 3.3), grâce auquel il se connecte sur Internet et envoie des requêtes format HTTP de la forme *GET* (cf. 3.1). Notre but a été ici de raccourcir au maximum les chaînes de caractères qui sont ainsi transmises pour minimiser le trafic et donc les coûts. Le serveur, sous la forme d'une servlet hébergée par un serveur Tomcat, fonctionne à la manière d'un serveur HTTP (cf. 3.1.4 et 4.3) ce qui contraint le format des requêtes décrit plus haut.

Coordinateur → Serveur

Le coordinateur possède en temps normal un accès à Internet de type ADSL (WiFi ou non). Si ce n'est pas le cas, une possibilité lui est offerte de se connecter via un téléphone portable qui fait office de modem ou par une carte

réseau GPRS ou UMTS (PCMCIA). Ici aussi la communication se fait à l'image des requêtes HTTP de la forme *POST* (cf. 3.1).

Dans les deux cas, les requêtes et leurs résultats sont des chaînes de caractères et donc sont totalement visibles sur le réseau. Les communications ne sont pas, dans cette version, sécurisées (cf. chap. 8 : "Evolutions futures").

4.3 Le serveur

Nécessaire pour la communication entre les utilisateurs mobiles et le coordinateur, le serveur se compose d'une servlet HTTP tournant sur un serveur Apache Tomcat. Dans notre projet, et pour garder l'optique d'une utilisation simplifiée, nous avons opté pour de simples fichiers XML (eXtensible Markup Language) en guise de base de données. Ce type de document étant très léger et ayant une syntaxe bien définie, il est facile de les stocker sur un serveur afin d'avoir un accès rapide aux données lors de l'utilisation du programme.

La transmission de mises à jour aux utilisateurs mobiles s'effectue en retour des coordonnées qu'ils envoient pour signaler leur position. Le serveur garde en mémoire les transferts déjà effectués et ne renvoie que ce qui est modifié. Ainsi, nous limitons la bande passante nécessaire ainsi que les frais qui en découlent.

4.4 L'application coordinateur

L'interface coordinateur du projet est une fenêtre java comprenant 3 parties. La première contient les renseignements de toutes les identités qui sont actuellement gérées par le coordinateur (panneau des identités). La deuxième permet d'afficher les renseignements des missions d'1, 2 ou 4 identités (panneau des missions) et enfin la dernière contient la carte et les différents points mobiles s'il y en a (panneau de la carte). Dès son ouverture, le programme lance un thread qui communique avec le serveur afin de mettre à jour l'interface coordinateur en cas de changements dans la base de données (fichiers XML sur le serveur) ou d'envoi de

nouvelles coordonnées des utilisateurs mobiles. Ce thread communique avec le serveur à intervalle régulier, réglable dans le menu des options (cf. 4.4).

4.5 L'application mobile

L'application mobile est composée de différents écrans successifs dans lesquels il est possible de naviguer en utilisant les boutons directionnels ou le joystick ainsi que les boutons situés sous l'écran. Il faut obligatoirement s'authentifier pour avoir accès à l'ensemble de l'application. Pour que l'application fonctionne correctement, il est impératif de pouvoir transmettre les coordonnées du lieu où se trouve l'utilisateur. Il est donc nécessaire d'établir une connexion entre le téléphone mobile et un récepteur GPS. Pour ce faire, nous avons utilisé une librairie open-source en J2ME qui permet de créer une connexion Bluetooth puis de traiter les trames provenant du GPS. La position est ensuite transmise par GPRS toutes les x secondes (à choisir dans les options).

Chapitre 5 : Manuels d'utilisation

5.1 L'interface coordinateur

Pour ouvrir l'application, il suffit de double-cliquer sur le fichier *DispatchLocator.jar* contenu dans l'archive qui se situe à l'adresse www.dispatch-locator.com.

La première étape dans le programme est de s'authentifier. Pour cela, une fenêtre demande à l'utilisateur d'entrer l'identificateur (ID) qui lui a été attribué lors de son inscription sur www.dispatch-locator.com. Il devra ensuite saisir l'adresse du serveur sur lequel sont stockés les fichiers nécessaires au fonctionnement de l'application (par défaut cette adresse est celle du serveur de l'Université de Genève) et saisir le port (par défaut 5000) (fig.12).



Fig.12 Fenêtre de login

Il est possible que deux types d'erreurs surviennent lors de l'identification. La première (fig.13.1) se produit lorsque le serveur ne fonctionne pas (le serveur Tomcat est éteint) ou lorsque le numéro de port n'est pas correct ; dans les deux cas la connexion avec le serveur ne peut se faire mais il arrive à le trouver sur Internet, l'adresse existe. La deuxième est plus concrète : le serveur n'existe tout simplement pas (fig.13.2). Le nom entré dans la fenêtre (fig.12) est faux.

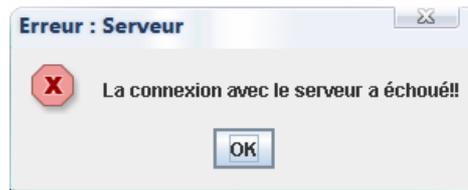


Fig.13.1 Problème sur le serveur ou erreur de port.



Fig.13.2 Mauvais nom de serveur.

Lorsque l'authentification se passe avec succès, une fenêtre s'ouvre afin de permettre de saisir le nom d'une carte (fig.14). Deux choix sont offerts : l'annulation qui ouvre un écran blanc qui peut être calibré quant bien même son utilisation sera moins précise, ou alors la saisie du nom d'une carte (sensible à la casse !) qui doit obligatoirement être contenue dans le dossier "map" qui se situe dans le même dossier que le *DispatchLocator.jar* qui permet de lancer le programme. Cette procédure permet aussi de retrouver automatiquement des fichiers de calibration contenus dans le dossier "calibration" qui se trouve au même endroit que le dossier "map".

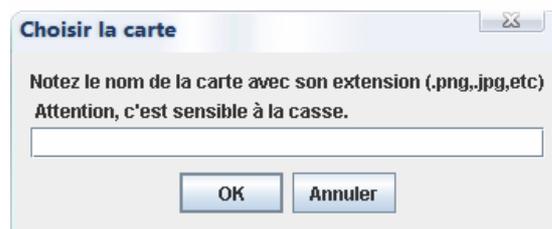


Fig.14 Choisir la carte

La fenêtre principale est composée de trois parties (fig.15) : la partie de droite qui permet de gérer tout ce qui concerne les identités des agents mobiles, la partie du bas qui permet de visualiser les missions des agents mobiles (4 agents à la fois

maximum) et pour finir la partie centrale qui gère la partie carte. Un clic droit sur l'un de ces composants donne automatiquement accès à un menu qui permet d'accomplir diverses actions. Nous y reviendrons plus en détails dans les sections suivantes.

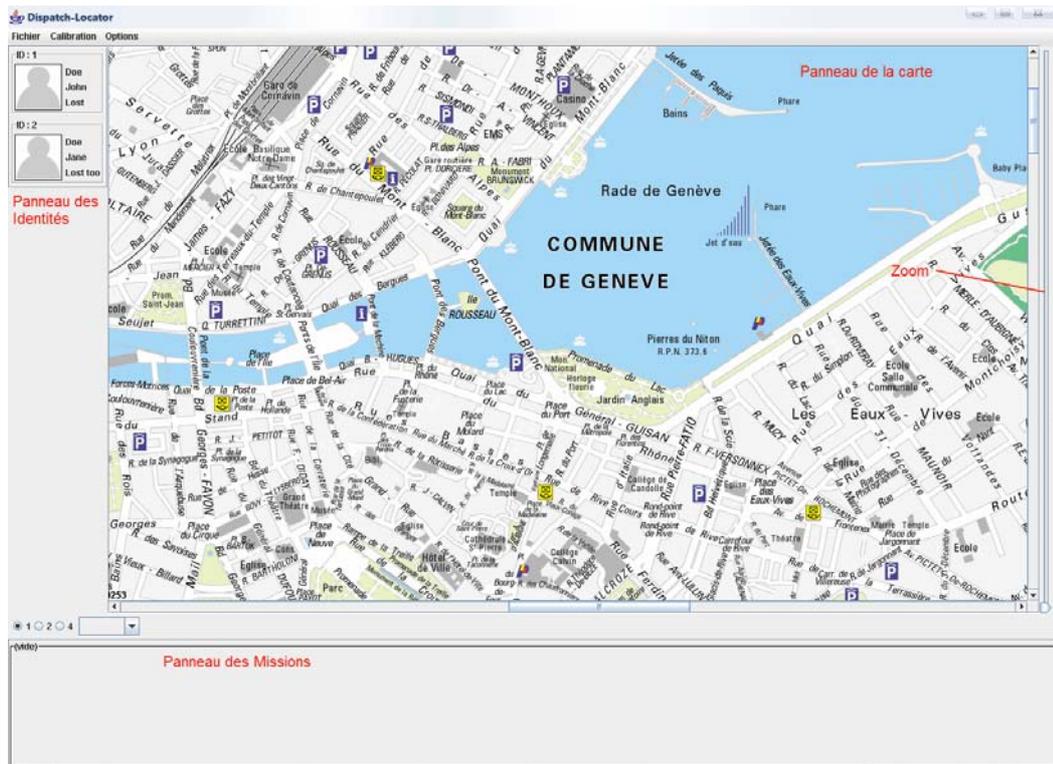


Fig. 15 Fenêtre interface coordinateur

Les menus :

A. Menu Fichier

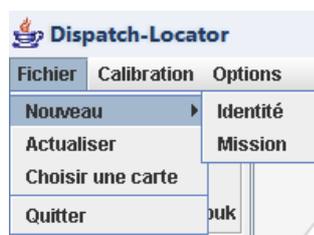


Fig. 16 Menu fichier

- Menu nouveau :

Ce menu permet de créer des missions et des nouvelles identités. Les missions créées par ce moyen prennent automatiquement les coordonnées $\langle 0 ; 0 \rangle$ car elles n'ont pas de rapport avec la carte et sont le plus souvent informatives. Pour créer une mission avec coordonnées, il suffit de faire un clic droit sur la carte, pour peu que celle-ci soit déjà calibrée. Reportez-vous à la section "*Panneau de la carte*" pour plus de précision.

- Menu Actualiser :

Permet de mettre à jour la fenêtre principale, les identités et les missions (fig.15).

- Choisir une carte :

Ouvre la même fenêtre qu'au démarrage de l'application après le login (fig.14) et permet de choisir une nouvelle carte tout en conservant les données en cours.

- Quitter :

Permet de quitter l'application.

B. Menu Calibration

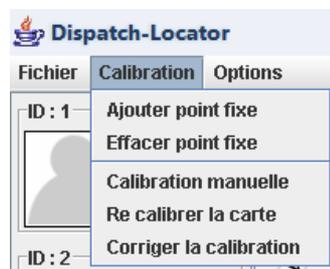


Fig. 17 Menu de Calibration de la carte

- Ajouter un point fixe :

Permet de rajouter sur la carte un point fixe (pour la définition voir la section "*Panneau de la carte*"). Si la carte n'est pas calibrée il suffit de poser deux points fixes précisément et la carte se calibrera toute seule. Pour se faire, le curseur sur la carte se transforme en viseur , permettant de cliquer sur l'endroit où poser le point fixe. Celui-ci apparaît alors comme un carré bleu à côté duquel se trouve le nom de ce qu'il représente. Pour annuler le viseur, il suffit de faire un clic droit sur la carte. Cette astuce est valable chaque fois que le viseur est visible sur la carte, un simple clic droit l'enlève.

- Effacer un point fixe :

Efface un point fixe de la carte. Si le point fixe était aussi un point de calibration, le point de calibration reste inchangé, il n'y a pas de perte de la calibration. Pour changer la calibration il faut aller dans le menu "*Corriger la calibration*" décrit plus bas.

- Calibration manuelle :

Ce menu permet de calibrer la carte à la main, c'est-à-dire d'entrer les coordonnées géographiques depuis le clavier (il est possible de trouver les coordonnées à différents endroits dont le programme Google Earth). Une fenêtre avertit l'utilisateur du début de la procédure (fig.18).

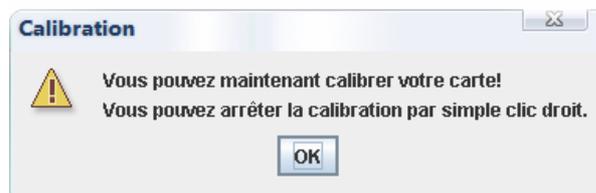


Fig. 18 Début de la calibration de la carte

Le curseur sur la carte se transforme en viseur  et il est dès lors possible de définir le premier point de calibration. Lors du clic une nouvelle fenêtre s'ouvre pour permettre la saisie manuelle des coordonnées géographiques (fig.19). Celles-ci doivent être soit au format WGS84 soit en coordonnées suisses CH1903. Il est impératif que les coordonnées envoyées par les utilisateurs mobiles correspondent à celles choisies pour calibrer la carte afin de garantir une correspondance des lieux entre les appareils.

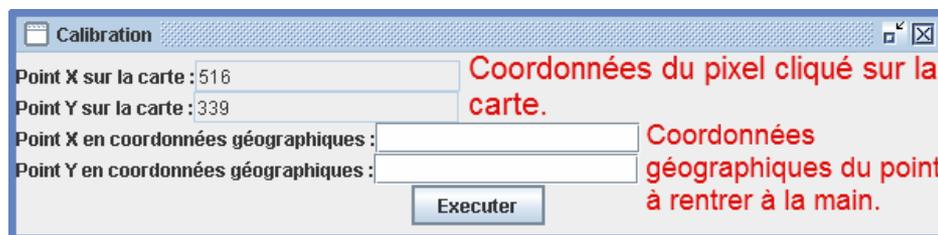


Fig. 19 Saisie manuelle des coordonnées

Un clic sur *Executer* enregistre le premier point et la procédure recommence pour le deuxième point.

Une fois les deux points correctement entrés, la carte est calibrée.

- Re-calibrer la carte :

Cette action réinitialise la calibration de la carte et le fichier de calibration attaché à celle-ci sera perdu. La suite de l'opération se passe exactement comme pour la "calibration manuelle".

- Corriger la calibration :

Une fois que la carte est calibrée, il se peut que la calibration soit imprécise. Il est alors possible de la corriger. Ce menu ouvre une fenêtre (fig.20) qui affiche les deux points de calibrations actuels. Il suffit de changer les valeurs de ces deux points pour corriger la calibration et ainsi obtenir une carte plus précise. Pour ce faire, l'utilisateur a le choix entre deux types de coordonnées terrestres : les coordonnées WSG 84 qui sont

une représentation décimale du système heure : minutes : secondes ou les coordonnées CH 1903 qui sont les coordonnées en mètre de la Suisse (Berne = 200'000 / 600'000). Ces coordonnées risquent d'être plus parlantes!

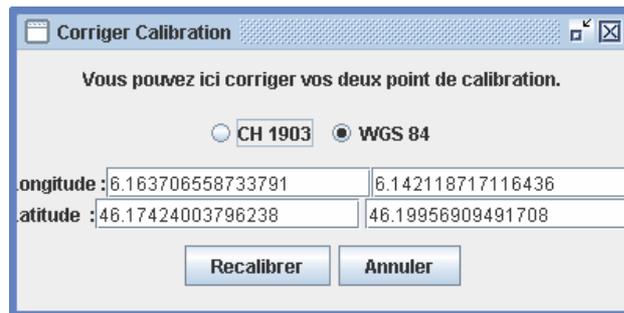


Fig.20 Correction de la calibration de la carte

C. Menu Options

- Taux d'actualisation :

Permet de modifier la fréquence de mise à jour de l'interface coordinateur (voir chap. 4 : "Architecture") qui s'effectue par défaut toutes les 5 secondes. (fig.21).

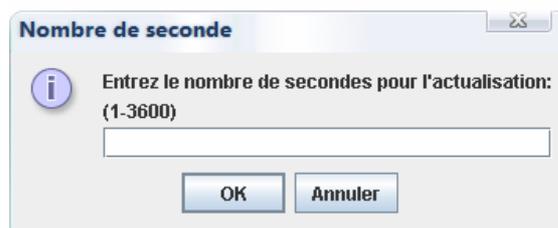


Fig.21 Nombre de secondes entre chaque mise à jour de l'application coordinateur

Les panneaux :

A. Panneau des identités :



Fig.22 Panneau des identités

Ce panneau, comme son nom l'indique, regroupe toutes les identités qui sont gérées par le coordinateur (fig.22). Elles doivent toutes avoir un nom, un prénom et un pseudonyme, et servent à authentifier les utilisateurs mobiles lors de leur connexion. Par un simple clic droit sur ce panneau, on accède aux commandes *ajouter*, *supprimer* ou *modifier* d'une des ces identités. Les identités en gris sont celles qui sont inactives, celles en orange sont en pause, celles en vert sont actives et en attente de missions et celles en rouge sont occupées, elles peuvent recevoir une mission mais en ont déjà une en cours. Les identités reviennent en gris après 5 min d'inactivité (pour l'instant pas réglable, cf. chap. 8 : "Evolutions futures").

B. Panneau des missions :



Fig.23 Panneau des missions

Le panneau des missions permet de visualiser les missions en cours pour chaque utilisateur (fig.23). On peut choisir d'afficher 1, 2 ou 4 utilisateurs à la fois selon les besoins. Les missions s'affichent les unes en dessous des autres et sont elles aussi colorées selon leur état : vert si la mission a été acceptée et que la personne est en train de l'exécuter, blanc s'il arrive que la mission ne soit attribuée à personne (très rare en pratique) et orange si la mission est en attente d'une réaction de l'utilisateur mobile. Les missions supprimées n'apparaissent plus dans ce panneau mais restent dans le fichier XML (cf. 3.1.4.B). En faisant un clic droit sur une mission, on peut la modifier, la supprimer mais aussi avoir accès à une fonction à utiliser avec parcimonie : *forcer l'effacement*. En effet cette dernière efface la mission de l'affichage mais sans prévenir l'utilisateur mobile de ce changement d'état ! Elle est là en cas de perte de réseau de l'utilisateur mobile ou fausse manœuvre de celui-ci, afin de corriger l'affichage.

C. Panneau de la carte :



Fig.24 Panneau de la carte

Ce dernier panneau contient la carte du programme choisie en arrivant dans l'application ou par la suite (fig.14 et 24). Comme les autres panneaux, celui-ci réagit au clic droit et donne accès à des menus. Si la carte est calibrée, il est possible de créer une nouvelle mission, sinon un message indique qu'aucune fonctionnalité n'est active pour le moment car la carte n'est pas calibrée. Les clients mobiles sont représentés sur la carte par des ronds rouges frappés d'un numéro correspondant au numéro de l'identité concernée. Si la carte n'est pas calibrée, les points rouges vont s'empiler dans le coin supérieur gauche de la carte et s'afficher les uns en dessous des autres. Autrement, les points s'affichent à l'endroit des dernières coordonnées reçues même si cela fait plus de 5 minutes que l'utilisateur n'a rien envoyé, que ce soit à cause d'une perte du réseau GPRS ou de réception GPS. Si l'utilisateur mobile est bien connecté, il est alors possible de lui envoyer un "GoTo" par simple clic droit sur le rond rouge de la carte qui lui

correspond. Chaque fois que l'utilisateur doit cliquer à un endroit précis sur la carte, le curseur se transforme en viseur . Ceci est valable pour les actions de calibration, de nouvelle mission et de Goto. Des points fixes peuvent également être affichés sur la carte. Ils peuvent être utilisés pour notifier des endroits fixes comme les tentes de secours, les *postes* de police ou tout autre lieu qui n'est pas destiné à bouger durant la manifestation. Ils servent également à calibrer la carte.

Les mises à jour

L'interface se met automatiquement à jour grâce à un thread qui tourne en arrière plan et qui est lancé à l'ouverture de l'application et arrêté à sa fermeture. Ce thread communique directement avec le serveur qui lui dit s'il doit lancer ou non un rafraîchissement des informations contenues dans les panneaux et surtout, afin de minimiser le trafic réseau, quel panneau il doit rafraîchir. Le thread demande aussi les mises à jour au niveau des coordonnées des utilisateurs mobiles (les points fixes sont stockés localement) pour accomplir leur réaffichage sur la carte. Il est possible via les menus de changer la fréquence des mises à jour (cf. "*Menu Options*").

5.2 Le mobile

Le meilleur moyen d'installer une application Java sur un téléphone mobile est de la télécharger depuis Internet. Ainsi, l'application est automatiquement installée dans le bon dossier et pourra être correctement exécutée. Il est également possible d'envoyer l'application via un câble USB, une connexion Bluetooth ou infrarouge mais certaines marques de téléphones mobiles (p. ex. Nokia et Samsung) ne permettent pas de placer par ce biais l'application dans un dossier depuis lequel elle pourra être exécutée.

Avant de lancer l'application pour la première fois, il faut s'assurer que le profil GPRS de l'opérateur soit bien configuré sur le téléphone. Si ce n'est pas le cas, il est possible de les obtenir gratuitement depuis les sites Internet des opérateurs. De plus, avant chaque utilisation, il faut veiller à ce que le Bluetooth soit bien activé.

Afin d'avoir accès aux fonctionnalités principales de l'application, une authentification est nécessaire (fig.25). A chaque fois, les informations saisies sont enregistrées dans la mémoire du téléphone pour éviter à la personne d'avoir à les re-saisir au prochain démarrage de l'application. Au moment de valider, en fonction de la configuration du téléphone mobile, une alerte peut apparaître pour demander la confirmation d'envoi d'informations sur Internet. Il faut impérativement accepter, sans quoi l'application ne peut pas communiquer avec le serveur.



Fig.25 Authentication

Si l'authentification est réussie, le menu principal de l'application s'affiche (fig.26). Dans le cas où les identifiants ne sont pas reconnus ou si la connexion au réseau ne peut être établie, un message d'erreur apparaît (fig.27).

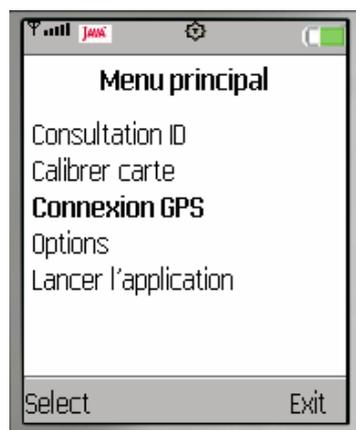


Fig.26 Menu principal

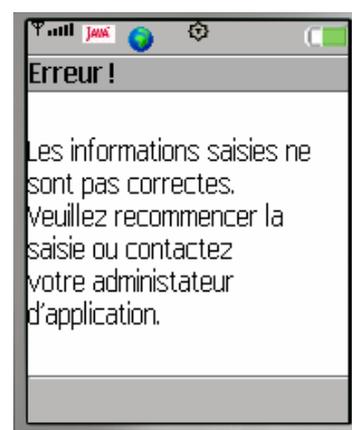


Fig.27 Authentication non-valide

Détails du menu principal :

- Consultation ID :

Affiche les informations de login de l'utilisateur ainsi que l'ID qui lui est attribué.

- Calibrer carte :

Permet d'envoyer les coordonnées de l'endroit où se trouve l'utilisateur, accompagnées d'une adresse ou d'une description afin de permettre au coordinateur de placer un point fixe sur sa carte (cf. 5.1 "Ajouter un point fixe") (fig.28).

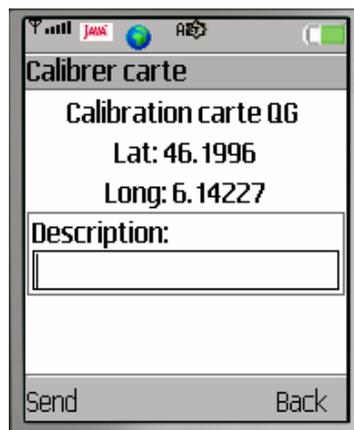


Fig.28 Ecran de calibration

- Connexion GPS :

Permet d'établir la connexion Bluetooth avec le récepteur GPS. Si le dernier récepteur à avoir été connecté est détecté, la connexion est automatiquement établie. Autrement, une liste présentant tous les appareils détectés s'affiche (fig.29a) et il suffit de choisir le nom correspondant au GPS pour que la connexion s'établisse (fig.29b). Dans le cas où une

connexion est déjà en cours, un message demande à l'utilisateur s'il souhaite se déconnecter (fig.29c).

Tout comme pour la connexion Internet, il est possible qu'une alerte s'affiche pour demander s'il faut autoriser la connexion Bluetooth. Il faut bien évidemment accepter.

NB. Il est préférable d'attendre que le GPS ait un FIX (possibilité de calculer sa position) avant de le connecter.



Fig.29a

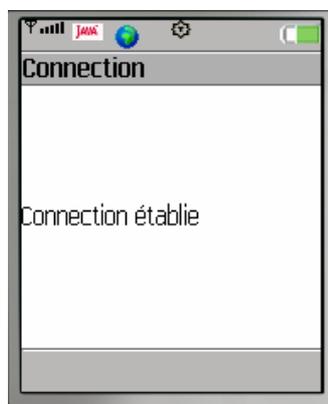


Fig.29b

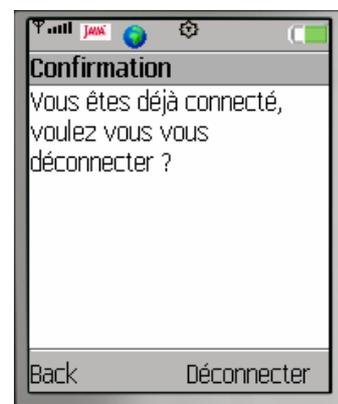


Fig.29c

Etapas pour la connexion avec le GPS

▪ Options :

L'utilisateur peut choisir ici (fig.30) :

- L'unité de distance : mètres
kilomètres
feet (pieds)
miles (environ 1,602 mètres)

- l'unité de vitesse : mètres / seconde
kilomètres / heure
feet / seconde
miles / hour (heure)

- le délai d'envoi des coordonnées : par défaut chaque 15 secondes et au maximum chaque 45 secondes
- le délai de rétroéclairage qui permet de maintenir l'écran allumé car sur certains modèles, la connexion Bluetooth avec le GPS pourrait s'interrompre lors de la mise en veille (provoque une plus grande consommation de la batterie).



Fig. 30 Ecran d'options

- Lancer l'application :

Affiche l'écran principal de l'application et démarre l'envoi des premières coordonnées chaque x secondes (spécifié dans les options).

L'écran principal :

Si l'utilisateur n'a pas encore de mission, l'écran affiche le statut de la personne, c'est-à-dire si elle est disponible pour une mission, si elle est déjà occupée par une mission qui ne provient pas du coordinateur ou si elle est en pause (fig.31 abc). Pour changer de statut, il suffit de se mettre sur le statut désiré puis d'appuyer sur *Change*. Le changement de statut est aussitôt signalé au coordinateur qui peut en fonction, décider d'attribuer ou non de nouvelles missions (cf. 5.1).

Si le GPS n'est pas connecté (fig. 31a) ou s'il n'a pas de FIX (fig.31b), un message s'affiche à l'écran. Il est tout à fait possible d'utiliser les statuts en présence de l'un de ces deux messages par contre la position n'est pas signalée au coordinateur et il n'est pas possible de recevoir de mission.

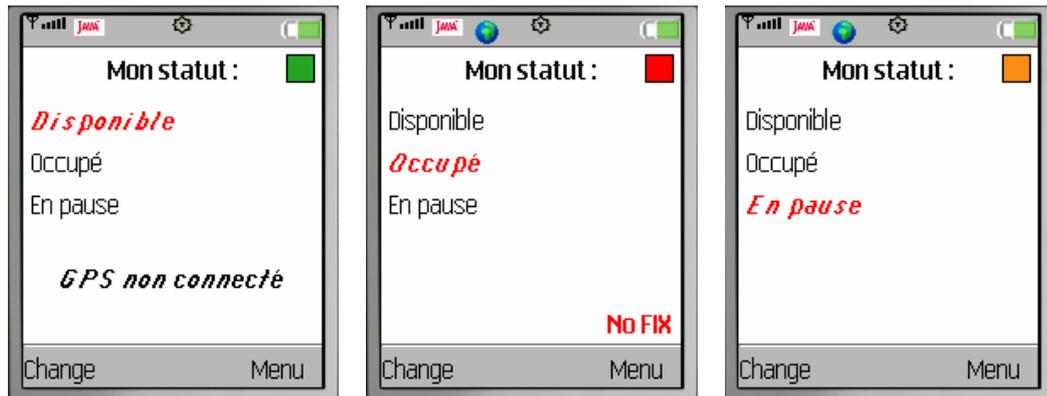


Fig. 31a

Fig. 31b

Fig. 31c

Les différents statuts et messages d'erreurs possibles

Des missions peuvent être reçues et il est possible de les *Accepter* ou de les *Refuser*. En cas de refus, le coordinateur peut éventuellement décider de réassigner la mission à la même personne. Il en existe trois types :

- une mission "standard", qui comporte un titre, une description ainsi que les coordonnées de la mission (l'endroit où se situe l'événement) (fig.32a).
- une mission "GoTo" qui comporte principalement les coordonnées de l'endroit où le coordinateur souhaite que l'utilisateur sur le terrain se rende (fig.32b).
- une mission "informative" qui comporte un titre, une description mais pas de coordonnées de destination (fig.32c).

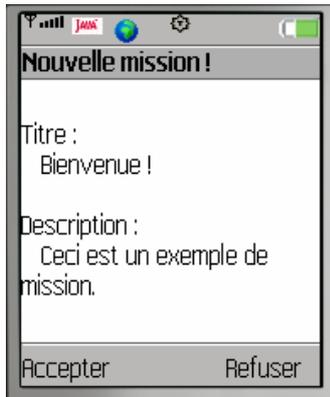


Fig. 32a Mission
"standard"

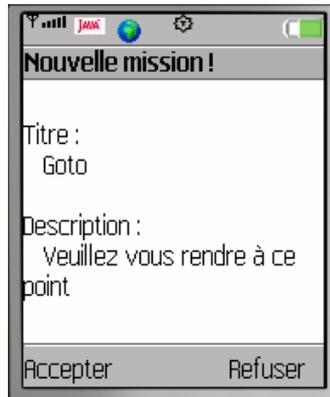


Fig. 32b Mission "GoTo"

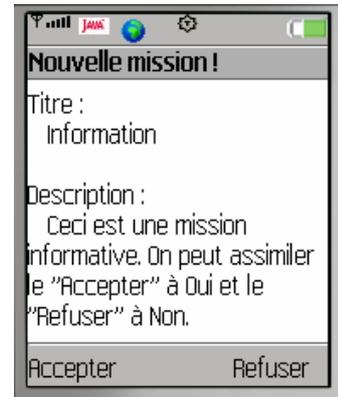


Fig. 32c Mission
"informative"

Si une mission est acceptée, elle est ajoutée à la liste des missions. Si c'est la première reçue, elle est automatiquement mise au statut "en cours". Dès lors, l'écran principal se présente comme suit pour les missions "standard" et "GoTo" (fig.33) :

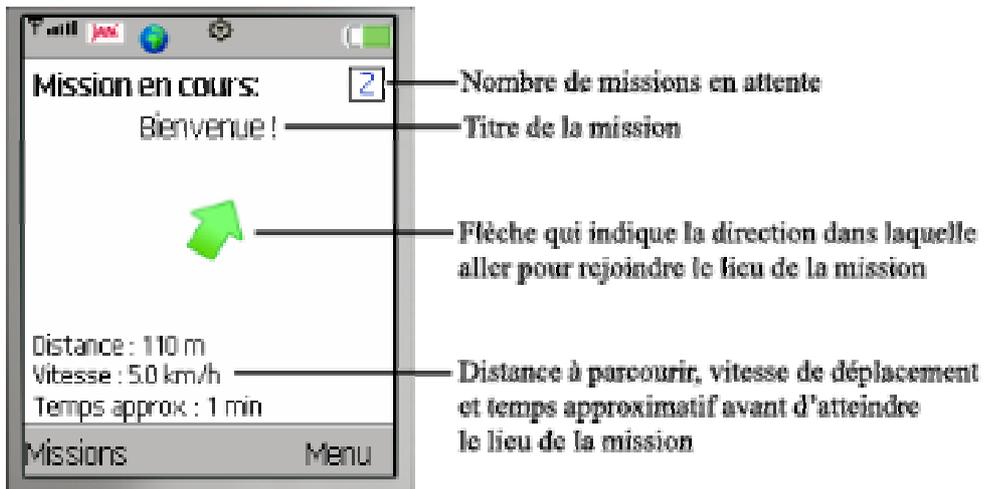


Fig. 33 Détails de l'écran principal

En revanche, si la mission en cours est de type "informative", seul la description de la mission est affichée (fig.34) :

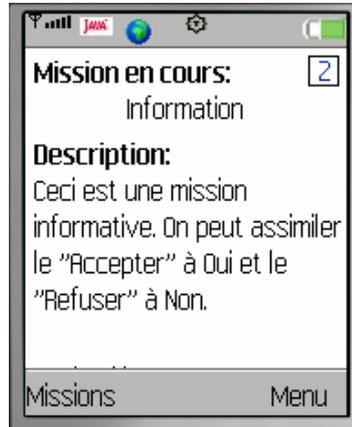


Fig.34 Ecran principal : mission "informative"

L'affichage principal possède trois menus :

- Missions :

Affiche la liste de toutes les missions reçues et non achevées (fig.35a). Pour afficher les détails d'une mission et en changer le statut, il faut sélectionner la mission désirée puis cliquer sur *Détails* (fig.35b). Pour accéder aux détails de la mission qui est en cours, il suffit d'appuyer une fois vers le bas sur le pavé directionnel.

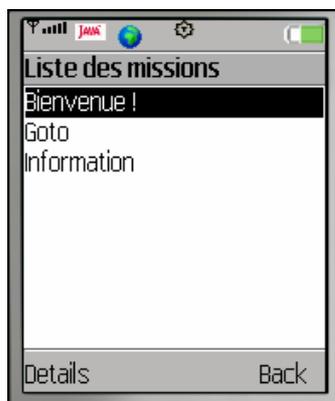


Fig.35a Liste des missions

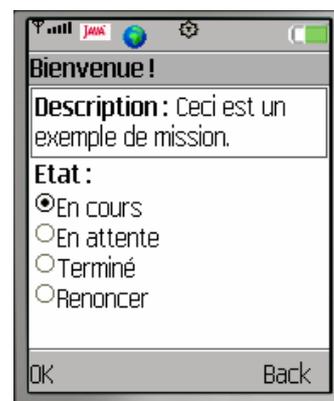
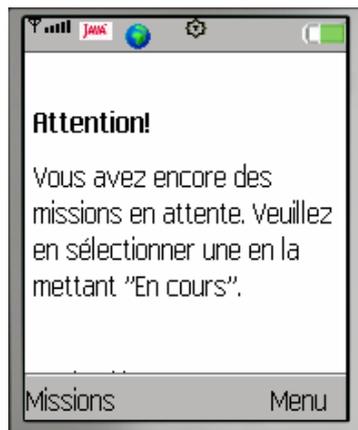
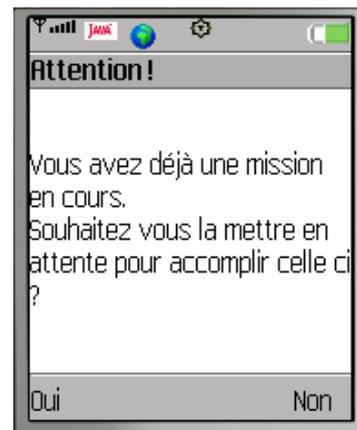


Fig.35b Détails d'une mission

Il existe deux situations particulières concernant les statuts des missions. La première est celle où toutes les missions sont mises en attente (l'utilisateur n'en exécute aucune) (fig.36a) et la seconde celle où l'utilisateur a déjà une mission en cours et veut en mettre une seconde en cours, ce qui est interdit (fig.36b).



*Fig. 36a Aucune mission
"En cours"*



*Fig. 36b Confirmation de
changement de mission*

Si les statuts "Terminé" ou "Renoncer" sont sélectionnés, la mission est supprimée de la liste.

- **Info GPS :**

Affiche un certain nombre d'informations relatives aux données reçues par le GPS.

- **Back :**

Ramène au menu principal et met en pause l'envoi des coordonnées au serveur.

Chapitre 6 : Cas d'utilisation – Communication tripartite

Les schémas ci-dessous décrivent la portée des actions (obligatoires ou possibles) qui peuvent être effectuées soit par le coordinateur soit par un utilisateur mobile. Toutes les actions transitent par le serveur qui fait office de relais entre tous les acteurs.

Le premier schéma montre les étapes qui découlent de l'ouverture des deux applications et le second correspond au cycle de vie d'une mission.

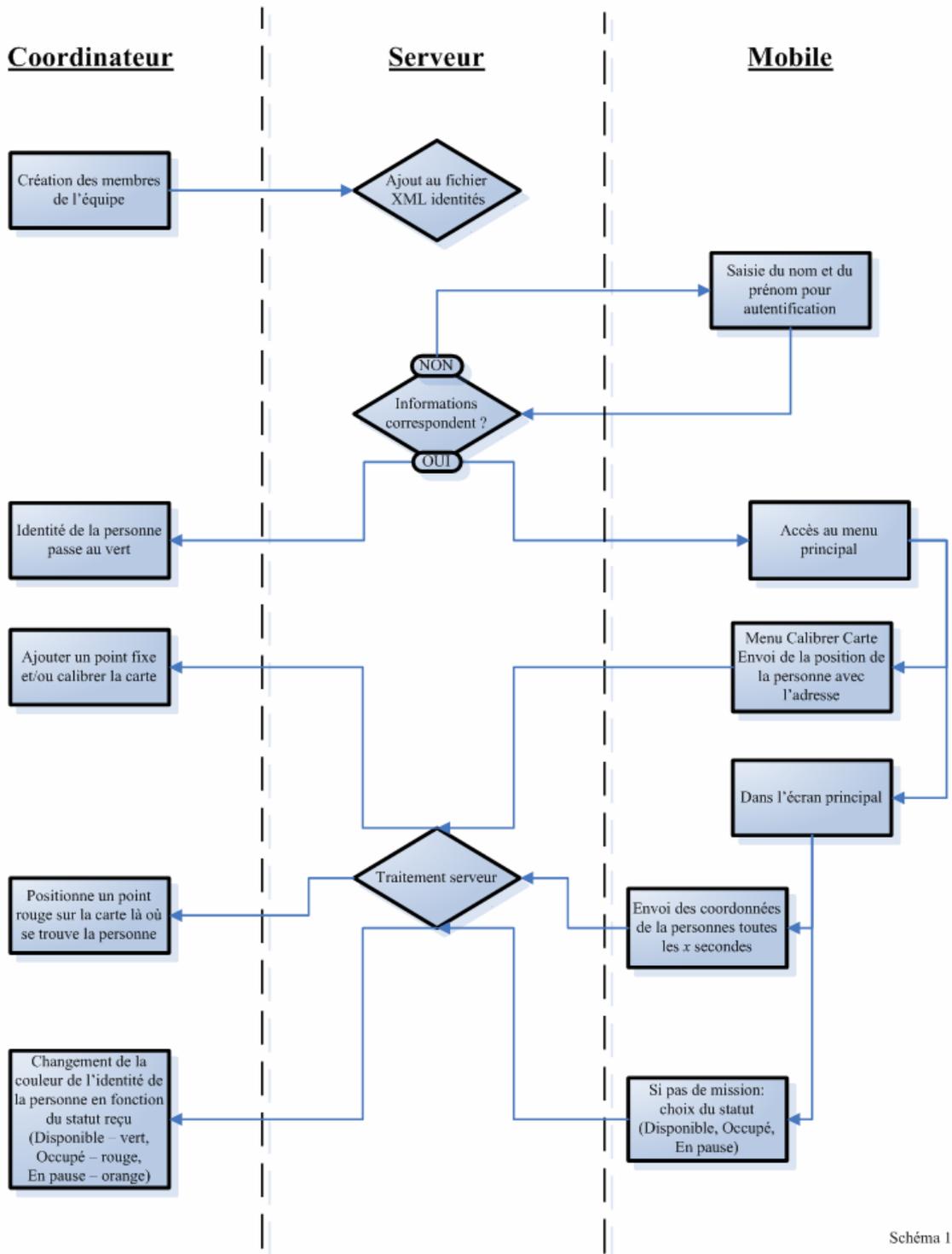


Schéma 1

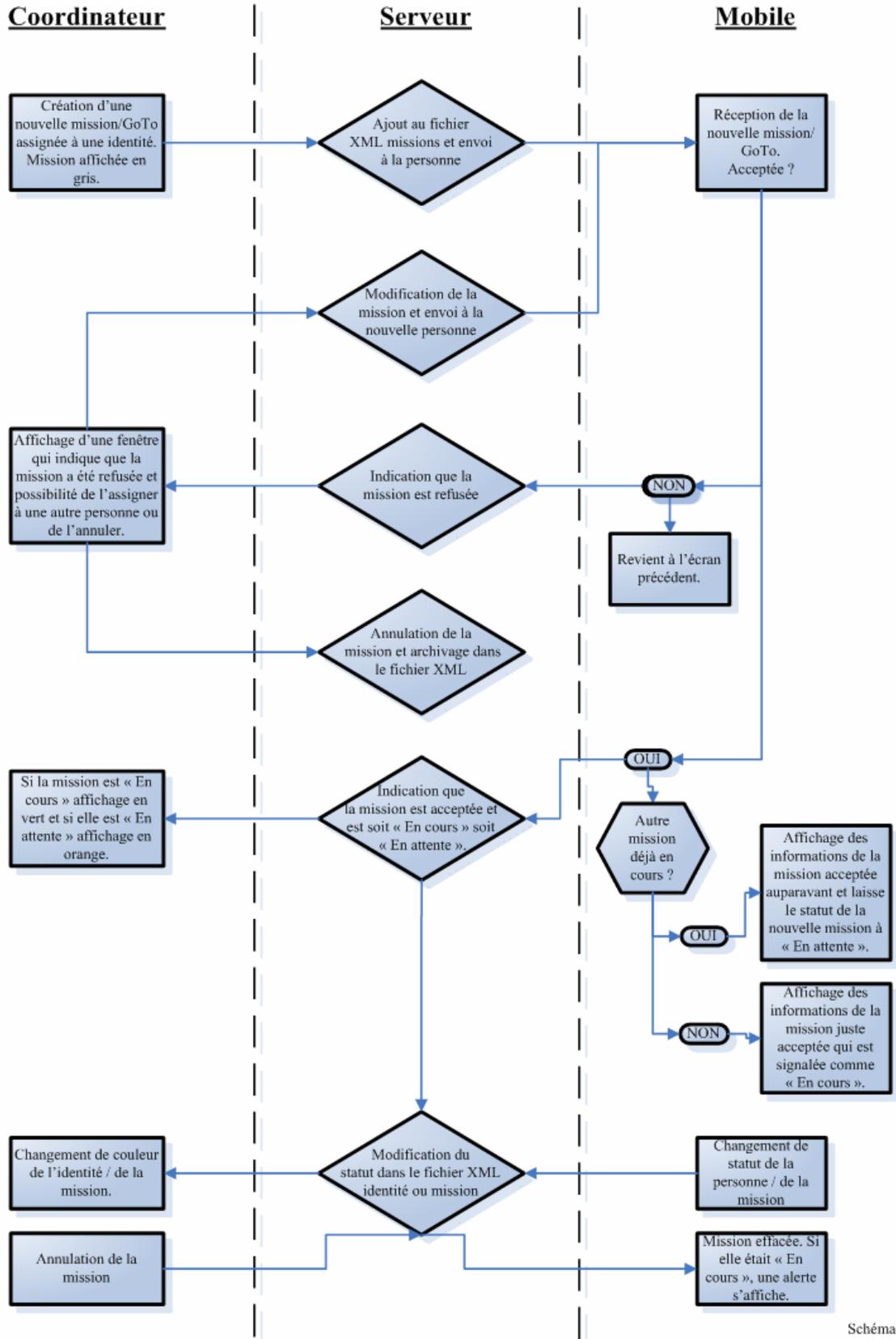


Schéma 2

Chapitre 7 : Collaboration avec les samaritains



Il y a près de 45'000 samaritains, tous bénévoles, répartis dans les quelques 1300 sections de Suisse. Ils sont chargés de prodiguer les premiers secours lors de manifestation avant l'intervention éventuelle d'un médecin ou d'une ambulance. La majorité d'entre eux ne sont pas des professionnels de la santé mais on suivi une solide formation dans le domaine.

A Genève, il existe 12 sections réparties dans les 45 communes chapeautées par une association cantonale AGSS (Association Genevoise des Section de Samaritains). Lors de grandes manifestations qui nécessitent des moyens humains supérieurs à ceux qu'une seule section pourrait fournir, la Commission Cantonale des Services Sanitaires de l'AGSS prend le relais, ce qui a été le cas pour la Lake Parade et les Fêtes de Genève.

7.1 Observation : Lake Parade

Lors de la Lake Parade 2006, nous sommes allés observer le travail des samaritains depuis le "central" où se trouve le coordinateur de l'ensemble des équipes présentes sur le terrain. Le dispositif prévu pour la manifestation était d'une vingtaine de samaritains qui fonctionnent par équipes de deux sur le terrain, ainsi que de deux autres en poste au central.

Le premier constat que nous avons fait est que les samaritains communiquent essentiellement par radio et que tout est consigné manuellement au central dans une main courante. L'informatique n'est pas du tout exploitée. Le coordinateur devait évaluer approximativement la position de chaque équipe principalement en fonction des informations sur l'avancée du cortège et de la position qui leur avait été attribuée au préalable dans le cortège. Il est arrivé que des samaritains qui devaient préciser leur position, aient du mal à repérer les noms de rues afin de

pouvoir fournir l'information. De même, il n'est pas toujours évident d'envoyer une personne à une adresse qu'elle ne connaît pas.

Un exemple de changement de tactique dû à cet inconvénient :

La Lake Sensation est la fête qui prolonge la Lake Parade. Les chars s'arrêtent et se stationnent sur le quai Gustave-Ador entre le Parc de La Grange et La Nautique. Le long de ce tronçon, il y a très peu de repères qui permettent d'indiquer précisément l'endroit où l'on se trouve. L'année passée, les samaritains étaient répartis sur l'ensemble du site et lorsque le coordinateur, cette fois dans un *poste* samaritain situé à proximité de la manifestation, était informé d'une urgence par les agents de sécurité, il n'était pas toujours évident pour lui d'envoyer l'équipe la plus proche du lieu d'intervention.

Cette année, ils ont décidé que tous les samaritains resteraient aux alentours du poste et que s'ils partaient en intervention, ils y reviendraient une fois leur tâche accomplie.

Ceci nous a conforté dans l'idée que notre système permettra très clairement d'éviter ce genre de situations où les samaritains ne peuvent être totalement déployés sur le terrain faute de pouvoir les localiser. Ainsi, la position et le statut de chaque équipe sont en tout temps affichés sur une carte et il est très facile de couvrir l'ensemble du terrain pour minimiser les temps d'intervention. De plus, grâce à la flèche de direction qui s'affiche sur le terminal mobile, les problèmes d'orientations sont réduits.

Nous avons également constaté que le rapport vocal via radio entre l'équipe samaritaine sur le terrain et le coordinateur au central reste indispensable. En effet, rien ne peut remplacer le questionnement direct, principalement lorsqu'il s'agit de déterminer s'il y a besoin d'appeler un médecin ou une ambulance. Rappelons que les samaritains ne sont pas, en majorité, des professionnels du domaine. L'interaction vocale permet également au coordinateur de percevoir par exemple le stress ou l'inquiétude dans la voix du samaritain, ce qui peut indiquer au coordinateur qu'il est en situation inconfortable voire en danger.

Grâce à notre système, la radio est libérée des communications non essentielles telles que le signalement de la position ou la demande d'informations d'orientation, ce qui libère la bande passante pour des transmissions plus importantes.

Enfin, ces quelques heures d'observation nous ont permis de mieux cerner quelles sont les fonctionnalités primordiales dont ils auront besoin et celles qui leur seront inutiles, afin d'ajuster notre travail en vue des Fêtes de Genève.

7.2 Déploiement : Fêtes de Genève

Lors des Fêtes de Genève, nous sommes retournés chez les samaritains afin de réaliser des tests sur le terrain et de leur montrer et expliquer en quoi consiste notre projet et comment il peut leur servir.

Leur organisation était la suivante (fig.37) : 5 postes samaritains étaient répartis tout autour de la rade et en moyenne 15 samaritains y ont assuré une permanence. En règle générale, les samaritains restaient à leur poste jusqu'à ce qu'ils soient envoyés par le coordinateur sur une mission et y revenaient une fois celle-ci achevée. De temps en temps des équipes partaient en patrouille.

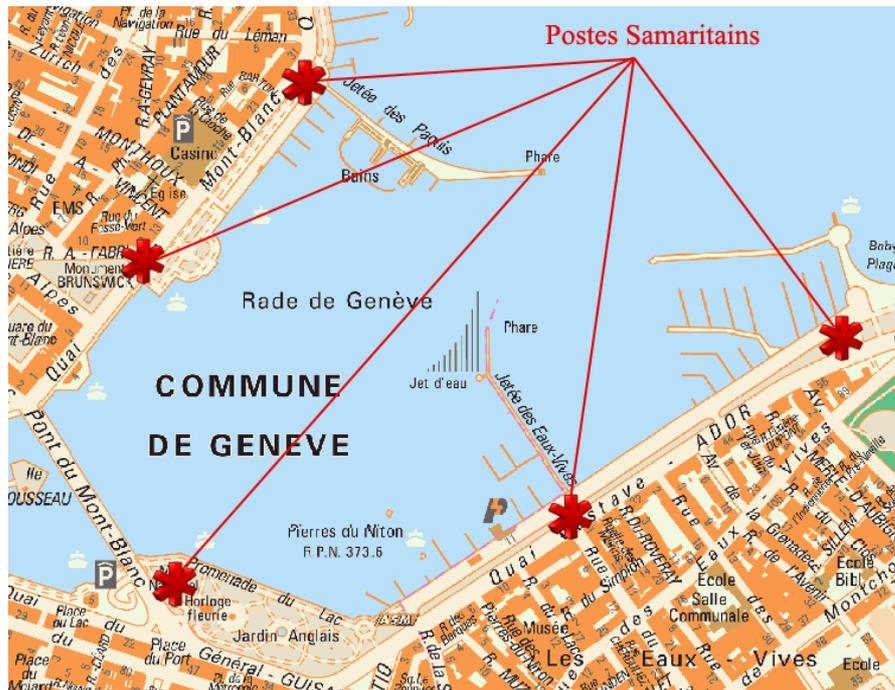


Fig. 37 Plan des postes samaritains

Il y a plusieurs réalités qui se sont révélées sur le terrain. La première est qu'il est très difficile de faire essayer une application à des personnes qui sont en train de travailler et qui peuvent à tout moment devoir partir en intervention. L'autre difficulté a été de trouver des téléphones compatibles. En effet, tous les modèles n'ont pas les caractéristiques nécessaires au bon fonctionnement de l'application, comme une version récente du profil Java ou les connexions GPRS et Bluetooth.

De plus, nous avons découvert qu'avec certaines marques (p.ex Nokia ou Samsung), il est impossible d'exécuter une application Java si elle n'a pas été téléchargée via Internet. Or n'étant pas au courant de cette contrainte, nous avons prévu de transférer l'application sur le téléphone des samaritains via une liaison Bluetooth. De fait, nous n'avons pas pu essayer notre application sur certains téléphones qui possédaient pourtant tous les paramètres nécessaires.

Nous avons également pu constater que la librairie Bluetooth que nous avons utilisée pour établir la connexion entre l'appareil mobile et le GPS ne fonctionnait pas à tous les coups. L'accent sera mis sur l'amélioration de cette liaison pour la prochaine version.

L'immense point positif de notre présence sur le terrain a été la réception du projet. En effet, l'idée et le concept a beaucoup plu aux samaritains qui y ont vu un moyen d'améliorer à moindre frais leur condition et cadre de travail.

Chapitre 8 : Evolutions futures

La version 1.0 de Dispatch-Locator que nous proposons pour clore notre projet de Bachelor est déjà entièrement fonctionnelle. Nous prévoyons cependant de continuer à développer ce projet par la suite afin d'y apporter des améliorations et d'en augmenter les performances.

Pour le serveur

- Ajouter la possibilité de diffuser les informations à des "viewers" sans que cela ne génère de conflits. Le "viewer" sera un programme qui ressemblera à l'application coordinateur mais sans la possibilité d'interaction avec les utilisateurs mobiles (p.ex. envoi de missions).
- Ajouter la capacité à gérer plusieurs manifestations différentes sur le même serveur, c'est-à-dire avoir plusieurs fichiers d'identités et de missions ainsi que plusieurs "viewers" par manifestation.
- Créer une DTD pour les fichiers XML.
- Offrir la possibilité de télécharger les fichiers XML pour les sauvegarder localement ou les lire dans un programme de visualisation de fichier XML.

Application coordinateur

- Ajout d'options permettant de :
 - o Choisir les couleurs pour la coloration des missions et des identités.
 - o Choisir le temps maximum d'inactivité des mobiles avant que ceux-ci ne soient considérés comme déconnectés ou ayant perdu le réseau.
- Redimensionnement manuel des panneaux (missions, identités et carte) selon les besoins (ex: augmenter la taille du panneau des missions pour pouvoir voir l'historique des missions).
- Possibilité de centrer la carte sur une identité sélectionnée.

- Pour chaque identité, affichage de la mission courante (panneau des missions) en haut de la liste des missions afin d'être visible pour que le coordinateur soit toujours au courant de la mission en cours pour une identité donnée.
- Affichage de drapeaux qui indiqueront la destination des missions et des GoTo. Un menu d'option permettra de les activer/désactiver ainsi que de modifier leur forme et leur couleur.
- Création de l'option de téléchargement et visionnement des fichiers XML stockés sur le serveur.

Application mobile

- Amélioration de la connexion Bluetooth avec le récepteur GPS afin d'accroître la compatibilité avec les différents téléphones mobiles.
- Perfectionnement de la vitesse d'affichage.
- Accroissement de la rapidité de transmission des messages par GPRS et tests avec l'UMTS.
- Réalisation de tests d'inter-opérabilité sur les modèles de téléphones mobiles les plus courants afin de dresser un tableau de compatibilité et éventuellement création d'une version modifiée de l'application en fonction des spécifications de chacun de ces téléphones.
- Développement d'une version de Dispatch-Locator pour Windows Mobile afin d'être utilisable sur un PDA ou un smartphone. Ceci permettrait d'utiliser des terminaux qui intègrent déjà un GPS.

Globalement :

- Dans la version actuelle de Dispatch-Locator, les données transitent en clair sur le réseau. Une amélioration certaine de la sécurité sera de crypter les données et par la suite d'intégrer un système de certification pour être sûr qu'aucune personne étrangère au système ne puisse interférer.

- Dans un souci d'internationalisation et de distribution, une traduction des sources du programme ainsi que des interfaces (mobiles et coordinateur) en anglais est prévue.
- Notre système étant malléable, il est possible de s'adapter en fonction des besoins de chacun. Nous sommes évidemment ouverts à toutes suggestions qui pourraient contribuer à l'amélioration de Dispatch-Locator.

Chapitre 9 : Conclusion

Dispatch-Locator est un système simple, efficace et gratuit pour gérer des équipes sur le terrain. Le but est que des équipes d'interventions puissent être gérées à distance par un coordinateur. Celui-ci peut les repérer sur une carte grâce aux coordonnées qu'ils transmettent automatiquement depuis un téléphone mobile couplé à un récepteur GPS. Il peut également communiquer avec elles pour leur confier des missions ou voir leur état afin de savoir en tout temps, combien d'équipes sont disponibles. Les interfaces des applications pour le coordinateur et pour les téléphones mobiles sont simples, intuitives et donc facilement utilisables sans formation préalable. Un serveur sert de relais entre ces deux applications et permet de conserver une trace de tout ce qui a été effectué (fichiers XML). Les applications sont en tout temps téléchargeables sur notre site Internet et sont faciles à installer. De plus, la portabilité de Java offre une grande flexibilité.

A l'origine, l'idée était de développer un système pour aider les samaritains lors de grands événements. Nous nous sommes certes basés sur leurs besoins mais en essayant de rester le plus ouvert possible pour que notre système soit utilisable dans des contextes qui diffèrent de ceux des urgences et de la sécurité. Par exemple, des sociétés spécialisées dans les livraisons en deux-roues pourraient y trouver un avantage puisque l'infrastructure mobile est très compacte et peut tenir dans une poche. Dès lors, la position des livreurs serait en tout temps signalée au coordinateur et il serait facile de leur assigner de nouvelles missions à accomplir. Avoir une équipe bien répartie sur le terrain permet d'envoyer la personne la plus proche d'une destination et d'éviter les trajets inutiles.

La version actuelle (1.0) de Dispatch-Locator est déjà pleinement fonctionnelle et nous avons pu effectuer des tests sur le terrain avec les samaritains lors des Fêtes de Genève. Le système y a été reçu avec un vif enthousiasme ce qui nous a conforté dans l'idée qu'il y a une demande pour ce type d'application d'autant plus si elle est gratuite. En cela, notre projet est un succès ! Nous allons d'ailleurs

poursuivre son développement afin de proposer très bientôt une nouvelle version qui comportera des améliorations et des nouvelles fonctionnalités (cf. chap. 8 : "Evolutions futures"). Elle devrait d'ailleurs être mise en service lors de la nuit de la Saint-Sylvestre à Genève, dans le cadre d'une collaboration avec les samaritains.

Comme mentionné auparavant, une des prérogatives de Dispatch-Locator est qu'il est open-source et distribué sous une licence LGPL (Lesser General Public Licence)*. Ceci permet aux groupes qui souhaitent l'utiliser, d'adapter le code à leurs besoins et surtout de sécuriser l'application selon leurs critères. Par contre, nos droits d'auteurs sont préservés et il n'est donc pas possible de s'approprier notre travail.

Nous avons créé un site Internet sur lequel il est possible d'obtenir les applications pour le coordinateur et pour les utilisateurs mobiles ainsi que le présent rapport. Il nous servira également à informer les personnes intéressées des mises à jour ou des changements qui pourraient intervenir à propos de Dispatch-Locator. Une FAQ sera mise en ligne par la suite, en fonction des questions des utilisateurs et de leur nombre.

Si vous êtes intéressés par ce produit ou pour toute suggestion, n'hésitez pas à nous contacter ! Voici l'adresse du site : www.dispatch-locator.com

* <http://www.gnu.org/licenses/lgpl.html>

Remerciements

Un grand merci à M. Michel Deriaz pour son idée qui est à l'origine de notre projet, pour sa motivation et pour le suivi de notre travail, ainsi qu'au professeur Dimitri Konstantas qui a bien voulu prendre avec nous le pari de réaliser ce projet.

Un merci tout aussi grand à nos familles et ami(e)s proche(s) pour leur patience et leur soutien lors des moments difficiles et pour les absences répétées lors des tests du programme aux Fêtes de Genève.

Nous tenons également à remercier :

Les Samaritains de Genève et l'équipe CCSS (Commission Cantonale des Services Sanitaires) pour leur accueil, leur patience et leur enthousiasme, et tout particulièrement M. Adrian Gutknecht pour son soutien et sa disponibilité et grâce à qui nous avons pu effectuer les tests chez les samaritains. Sur le terrain, ce sont Véronique Cardeccia et Christophe Gallay dont l'aide a été précieuse ainsi que tous les samaritains qui ont bien voulu tester notre application.

A la centrale téléphonique du 144, MM. Alain Durant, responsable administratif, et André Kunzler, responsable technique, pour avoir mis à disposition les infrastructures nécessaires à nos tests et pour les nombreux sachets de thé qui n'ont pas manqués lors de nos longues nuits d'essais.

Toutes les personnes du forum du site www.developpez.com et une note spéciale pour Nicolas Cynober pour son excellent tutoriel sur JDOM, la librairie XML du projet.

M. Nicolas Mayencourt Ingénieur Système de l'Université de Genève pour ses précieux conseils et son aide lors du déploiement du serveur Apache sur les installations de l'Université.

A tous nos relecteurs qui de part leur lecture assidue et leurs critiques nous ont permis de finaliser ce rapport.

Joëlle et Stéphane

Références

Ouvrages :

LEMAY Laura, CADENHEAD Rogers, Le Programmeur Java 2, CampusPress 1999.

LI Sing, KUDSEN Jonathan, Beginning J2ME : From novices to professionnels, Third Edition, Apress 2005.

HUNTER Jason, CRAWFORD William, Servlets Java – Guide du programmeur, 2^{ème} édition, O'Reilly 2002.

Sites Internet :

<http://www.universal-locator.com>

<http://www.developpez.com>

<http://java.sun.com>

<http://www.jdom.org>

<http://www.gpspassion.com>