

Christophe Berger  
berger\_c@epita.fr  
EPITA Promotion 2007  
Spécialisation CSI

## Rapport de stage de fin d'études

# ANALYSE ET DÉVELOPPEMENT DE PROTOCOLES DE COMMUNICATION POUR UN RÉSEAU SOCIAL MOBILE

Stage au LIP6 sous la direction de Naceur Malouch

Février à Juillet 2007



LIP6 - Laboratoire d'informatique de Paris 6  
Département Réseaux et Systèmes Répartis  
104 avenue du Président Kennedy  
F-75016 Paris



EPITA - École Pour l'Informatique et les Techniques Avancées  
14-16 rue Voltaire  
F-94276 Le Kremlin-Bicêtre



# Résumé du stage

MoSoMuSo (Mobile Social Music Software) est un projet regroupant Sony Computer Science Laboratory, Clicmobile et le LIP6 et est financé par le réseau pour la Recherche et l'Innovation en Audiovisuel et Multimédia (RIAM).

MoSoMuSo crée une convergence entre écoute de musique, réseau mobile et réseau social. L'application doit permettre à terme de transporter un flux musical en y intégrant un contexte social et doit pouvoir fonctionner sur des appareils disponibles au grand public tels que téléphones mobiles ou assistants personnels équipés d'une interface WiFi et d'une interface au réseau mobile 3G. Le contenu musical se base sur un concept développé par Sony CSL, la musique malléable. Cela consiste à remixer en direct une piste musicale en utilisant des informations fournies par différents capteurs, ou ici, par les membres de la communautés qui participent à la création musicale. Chaque participant pouvant influencer la transformation de la musique en temps réel.

Le Laboratoire d'Informatique de Paris 6 (LIP6) regroupe les activités de recherche en informatique de l'Université Pierre et Marie Curie - Paris 6 (UPMC) et constitue une unité mixte de recherche (UMR 7606) de l'UPMC et du Centre National de la Recherche Scientifique (CNRS). Mon stage s'est déroulé au sein du département réseau et systèmes répartis dans le groupe de recherche Network and Performance Analysis (NPA).

Le groupe NPA se concentre sur les protocoles liés aux réseaux du futur, dont les protocoles sans fil dans le but de créer des réseaux mobiles autonomes, l'IPv6, les performances des réseaux...

Le sujet du stage consiste à concevoir et à implémenter la construction d'un réseau communautaire. Idéalement, cette tâche regroupe tous les mécanismes permettant la création physique du réseau en croisant à la fois la géolocalisation et le profil des membres de communautés afin de permettre la communication entre ceux-ci et d'optimiser le routage. Le but étant le transport de flux musical en temps réel entre les membres.

Le travail réalisé au cours du stage a été plus vaste et a surtout concerné les aspects d'ingénierie du projet. Il a commencé par le choix de la plateforme de test puis la création d'une interface graphique pour cette plateforme. Ensuite deux prototypes ont été développés afin de valider le concept. Le prototype implémente une composante essentielle de la construction du réseau communautaire qui est la boussole : un mécanisme incitatif qui a pour objectif de reconstruire le réseau. Des tests grandeur nature ont été réalisés avec des volontaires pour vérifier le schéma d'utilisation.

Aujourd'hui la première étape de prototypage est terminée et le travail s'oriente vers le routage dans les réseaux auto-organisés dans le but de connecter les membres de la communauté.



# Remerciements

Je tiens à remercier tout particulièrement Naceur Malouch qui m'a encadré durant mon stage ainsi que Guillaume Valadon qui m'a guidé et m'a permis d'entrer dans le projet plus facilement.

Les autres membres du projet m'ont également été d'une grande aide et d'un grand soutien, notamment Atau Tanaka (Sony CSL) que je remercie vivement pour son aide et ses conseils. Bénédicte Le Grand pour sa collaboration au projet et ses idées sur les mécanismes incitatifs et Alex Kummerman (Clicmobile) avec qui j'ai pu avoir de grandes discussions à propos du projet et qui m'ont donné un nombre impressionnant d'idées, je les en remercie.

Je tiens à remercier l'ensemble de l'équipe NPA du LIP6 qui m'a accueilli très chaleureusement, particulièrement les thésards.

---

# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Stage</b>                           | <b>15</b> |
| 1.1      | LIP6 [1]                               | 15        |
| 1.2      | Projet MoSoMuSo                        | 16        |
| 1.3      | Partenaires                            | 16        |
| 1.4      | Contexte du stage                      | 17        |
| <b>2</b> | <b>Applications et réseaux sociaux</b> | <b>23</b> |
| 2.1      | Applications sociales                  | 23        |
| 2.2      | Applications sociales musicales        | 24        |
| 2.3      | Musique mobile malléable               | 25        |
| <b>3</b> | <b>Technologies réseau</b>             | <b>27</b> |
| 3.1      | Techniques d'accès                     | 27        |
| 3.2      | Méthodes de positionnement             | 30        |
| 3.3      | Systèmes d'exploitation mobile         | 32        |
| 3.4      | Protocoles réseau                      | 33        |
| 3.5      | Protocoles réseau niveau transport     | 34        |
| <b>4</b> | <b>Analyse et conception</b>           | <b>35</b> |
| 4.1      | Fonctionnement général                 | 35        |
| 4.2      | Approches                              | 35        |
| 4.3      | Réseau social                          | 37        |
| 4.4      | Boussole et positionnement             | 37        |
| 4.5      | Connectivité réseau                    | 38        |
| 4.6      | Interface utilisateur                  | 39        |
| 4.7      | Scénario d'utilisation                 | 39        |
| 4.8      | Rafraîchissement serveur               | 40        |
| <b>5</b> | <b>Réalisation</b>                     | <b>45</b> |
| 5.1      | Choix techniques                       | 45        |
| 5.2      | L'application                          | 46        |
| 5.3      | Tests                                  | 49        |
| 5.4      | Dissémination                          | 51        |
| 5.5      | Problèmes rencontrés                   | 51        |
| <b>6</b> | <b>Bilan</b>                           | <b>55</b> |





# Table des figures

|     |   |    |
|-----|---|----|
| 1.1 | Déroulement du stage, diagramme de Gantt . . . . .  | 21 |
| 3.1 | Structure simplifiée d'un réseau GSM . . . . .      | 28 |
| 4.1 | Fonctionnement général de l'application . . . . .   | 36 |
| 4.2 | Esquisse de la boussole . . . . .                   | 38 |
| 4.3 | Fonctionnement logique de l'application . . . . .   | 40 |
| 4.4 | Modélisation de l'application (prototype) . . . . . | 41 |
| 5.1 | Interface graphique de l'application . . . . .      | 47 |
| 5.2 | Simulateur GPS . . . . .                            | 48 |



# Liste des tableaux

|     |   |    |
|-----|---|----|
| 3.1 | Récapitulatif des normes et débits . . . . .                          | 31 |
| 5.1 | Comparatif Apple iPhone, Nokia N95, HTC P3300 Premium . . . . .       | 46 |
| 5.2 | Comparatif de performances avec <i>ffbench</i> . . . . .              | 50 |
| 5.3 | Comparatif de performances avec <i>SPMARK</i> <sup>04</sup> . . . . . | 50 |
| 5.4 | Performance de la lecture mémoire (s) . . . . .                       | 51 |



# Introduction

L'être humain fonctionne au gré des rencontres et des opportunités qui s'offrent à lui, ainsi, les amis se trouvent souvent au hasard d'une promenade. L'informatique s'inspire beaucoup du comportement humain mais il en est encore un qui lui résiste c'est cette spontanéité. Les réseaux sont depuis longtemps des domaines fermés où tout se fait par câble enfermés et protégés ou par ondes radio chiffrées. Dans ce stage le but est de permettre les interactions spontanés entre les gens et ainsi leur faire découvrir de nouveaux horizons grâce aux réseaux sans fil.

Pour s'ouvrir sur les autres il n'y a rien de mieux que le formidable vecteur de communication qu'est la musique. Le projet sur lequel a porté mon stage de fin d'études d'EPITA est un projet à la convergence des réseaux et de la musique, MoSoMuSo (Mobile Social Music Software).

Mon stage a été réalisé au sein du Laboratoire d'Informatique de Paris 6 (LIP6) dans le cadre d'une collaboration entre le LIP6, Clicmobile et Sony Computer Science Laboratory (Sony CSL) et était destiné à développer les protocoles de communication pour une application de réseau social mobile intégrant un contenu musical.

J'ai pu prendre part à ce projet en qualité d'ingénieur chargé du développement réseau. De ce fait j'ai également intégré une équipe composée de partenaires très différents et intéressés par des aspects souvent contradictoires du projet.

Le projet MoSoMuSo doit proposer à terme une plateforme croisant à la fois des données géolocalisées, des profils de membres de communautés et permettant la manipulation temps réel de contenu afin de servir pour des travaux de recherche. La manipulation de contenu est un terme générique pour une application servant d'outil pour générer ou modifier des écrits, des sons ou de la vidéo. MoSoMuSo se basant plus particulièrement sur du contenu musical. En effet, le projet servira de plateforme d'expérimentation pour d'autres travaux de recherche touchant différents domaines tels que la mobilité, l'étude des usages des périphériques musicaux mobiles ou simplement les protocoles réseau mobiles.

Le concept réseau développé lors de ce stage est une approche différente des approches traditionnelles qui partent du principe que la connectivité est acquise ou alors qui utilise une connectivité globale du type 3G, nous avons développé un concept qui incite les gens à bouger en direction de zones de couverture pour obtenir la meilleure connectivité.

Le stage sera expliqué en détail dans une première partie qui décrira également le cadre du projet et mon environnement de travail. Un état de l'art sur les technologies musicales et sur les technologies réseau sera ensuite fait pour introduire les choix de conception qui m'ont permis de développer l'application qui a été finalisée à la fin de mon stage et qui est décrite dans le chapitre de réalisation.



# Chapitre 1

## Stage

L'offre concernait l'embauche d'un ingénieur de recherche et développement pour une durée d'un an au sein du Laboratoire d'Informatique de Paris 6 (**LIP6**).

Elle porte sur la conception et le développement de protocoles de communication pour Pocket PC et/ou téléphones mobiles et, plus particulièrement, sur la prise en charge du développement de techniques d'auto-configuration et d'auto-organisation. Dans un deuxième temps, sur l'étude du système d'exploitation du dispositif mobile et l'adaptation des protocoles conçus. Puis sur l'intégration du module réseau développé dans le système choisi. Pour se terminer sur la mise en place d'une plateforme de test et l'expérimentation des protocoles sur cette plateforme.

J'ai donc pris part au projet en tant qu'ingénieur réseau et j'ai été chargé du développement complet de l'application et de l'implémentation des protocoles qui seraient développés au sein de l'équipe pour le projet **MoSoMuSo**.

### 1.1 LIP6 [1]

Le LIP6 est un laboratoire de recherche sous tutelle de l'Université Pierre et Marie Curie, et du CNRS. Avec 128 chercheurs permanents et 231 doctorants, il est l'un des principaux laboratoires de recherche en informatique en France, et le plus gros laboratoire d'informatique de la région parisienne. Le laboratoire couvre un large spectre d'activités regroupées au sein de 5 départements : Calcul Scientifique ; Décision, Systèmes Intelligents et Recherche opérationnelle ; Données et Apprentissage Artificiel ; Réseaux et Systèmes Répartis ; Systèmes embarqués.

En complément de la recherche académique, le LIP6 a une longue tradition de coopération avec des partenaires industriels dans de très nombreux projets nationaux, européens ou internationaux. Deux centres R&D ont été créés : le CERME, Centre Européen de Recherche en Micro-Electronique sur les systèmes embarqués, et Euronetlab, sur l'Internet et les réseaux de télécommunication.

Le LIP6 est également impliqué dans les pôles de compétitivité de l'Ile-de-France : Cap Digital sur le contenu numérique et System@tic sur les systèmes embarqués. Il a également des équipes communes avec l'INRIA sur les thématiques du calcul formel et des systèmes répartis.

La coopération internationale est une constante pour les activités du laboratoire.

Le LIP6 est membre de plusieurs réseaux d'excellence et développe également des relations suivies avec des universités au Brésil, aux Etats-Unis, au Japon, en Tunisie et dans de nombreux pays européens. Le laboratoire est largement ouvert aux projets de coopération et à l'accueil de visiteurs scientifiques.

Le laboratoire est impliqué dans des enseignements liés à la recherche qui sont dispensés au Master « Sciences et technologie » à l’université Pierre et Marie Curie - Paris 6.

### Département “Réseaux”

Le département « Réseaux et Systèmes Répartis » se concentre sur la conception de solutions pour construire et gérer les réseaux et systèmes du futur. Il est constitué de quatre équipes : MoVe (Modélisation et Vérification), REGAL (Répartition et Gestion des Applications à Large Echelle), NPA (Networks and Performances Analysis) et PHARE.

### Équipe NPA

L’équipe NPA vise à développer une vision pour l’Internet du futur, ainsi que de concevoir des solutions pour le représenter et le contrôler. La cible de l’équipe est le contrôle des réseaux omniprésents, mobiles et dynamiques qui se développent partout dans nos environnements privés et professionnels. Le coeur de leur travail concerne des problèmes liés aux multimédia et aux réseaux mobiles, à la gestion de ressources, à l’extensibilité, aux réseaux ambiants, et à la gestion de réseaux de contenu. Par ailleurs, un travail significatif est développé autour de la mesure de l’Internet, de la modélisation et de l’ingénierie du trafic.

## 1.2 Projet MoSoMuSo

Le projet **MoSoMuSo** est un projet innovant conduit par trois partenaires et financé par le **RIAM**<sup>1</sup>. **MoSoMuSo** signifie “Mobile Social Music Software” et cherche à créer un logiciel mobile mettant en jeu une application musicale dans un contexte de réseau social.

Un réseau social peut être vu comme une carte des relations entre individus. Dans notre cas nous nous basons sur les relations d’amitié et de confiance afin de mettre en place une application de création musicale au sein de communautés. La finalité est d’embarquer le concept de musique malléable développé au sein du laboratoire d’informatique de Sony, Sony Computer Science Laboratory, Paris par Atau Tanaka<sup>2</sup>. Le concept et les travaux précédents sont présentés dans la partie 2.3. Il se base sur les interactions entre personnes afin de modifier l’ambiance sonore et ainsi faire participer les gens à une forme de création musicale. Ainsi chaque participant d’une *communauté* (un groupe d’amis ou de personnes intéressés par la même chose au même moment) peut prendre part à une expérience musicale à partir de son propre dispositif mobile, téléphone, assistant personnel ou ordinateur portable.

Les partenaires impliqués dans ce projet sont, d’une part Sony CSL par l’intermédiaire d’Atau Tanaka mais également le **LIP6** par le biais de l’équipe **NPA** qui apporte son expertise en terme de réseau et aussi “Clicmobile”<sup>3</sup>, qui apporte ses connaissances et son savoir faire dans le domaine des réseaux sociaux et des plateformes mobiles.

## 1.3 Partenaires

Voici une brève présentation des partenaires et les compétences qu’ils apportent au projet.

---

<sup>1</sup>Réseau pour la Recherche et l’Innovation en Audiovisuel et Multimédia, <http://www.riam.org>

<sup>2</sup><http://www.csl.sony.fr/General/People/StaffPage.php?username=atau>

<sup>3</sup><http://www.clicmobile.com>



### 1.3.1 Clicmobile

Clicmobile se positionne comme un éditeur d'applications mobiles, créatif sur le versant des usages qu'il observe et anticipe mais également sur le versant technologique avec la conception d'environnements de développement et d'applications reconnus innovant par le Ministère français de la Recherche en 2005. Les réalisations de CLICMOBILE sont exploitées par SFR et Orange. Clicmobile met à la disposition de ses clients une offre variée, composée d'une part d'un portefeuille propre de services et d'autre part d'applications réalisées pour des annonceurs dans le cadre d'animations événementielles permettant de développer les nouveaux usages du téléphone mobile (MMS, Vidéo, Musique 3G). Clicmobile a développé une forte capacité, distinctive sur le marché, dans le domaine de l'interactivité mobile/écran et de la manipulation des sources photos/vidéo/audio.

Au sein du projet, Clicmobile apporte ses compétences et son expérience sur les bases de données et sur les réseaux sociaux. Ils ont également une expertise en applications mobiles interactives. De plus, Clicmobile entretiens une relation privilégiée avec Nokia et divers opérateurs ce qui apporte également plus de crédibilité au projet.

### 1.3.2 Sony CSL

Le "Sony Computer Science Laboratory Paris" (Sony CSL Paris) est la branche internationale du "Sony Computer Science Laboratory" de Tokyo. Les deux laboratoires ont été créés par Sony afin de conduire des recherches fondamentales dans les domaines de l'informatique et des technologies. L'objectif de ces laboratoires est de contribuer activement au développement social et industriel à travers des recherches innovantes qui vont au-delà du *XXI<sup>e</sup>* siècle et qui ont le potentiel de révolutionner les techniques existantes dans les domaines des technologies.

Les laboratoires soutiennent des chercheurs en leur permettant de travailler librement sur les sujets qui les intéressent. Leurs résultats de tous types (logiciel, articles techniques, articles de conférence) sont publiés en leur propre nom.

Sony CSL apporte au projet sa grande expérience musicale et le concept de musique malléable.

## 1.4 Contexte du stage

Le stage s'est déroulé au sein de l'équipe **NPA** au **LIP6** dans les locaux provisoires Passy-Kennedy, 104 avenue du Président Kennedy, 75016 Paris.

Lors de mon arrivée le projet **MoSoMuSo** était à sa première année d'existence. Le travail déjà réalisé se résumait à diverses choses.

D'une part chez Sony CSL, Atau Tanaka avait implémenté une application fonctionnelle de musique malléable afin de la présenter pour les dix ans d'existence du CSL.

D'autre part, Clicmobile avait effectué un travail de fond sur les réseaux sociaux afin de définir exactement les besoins en terme de base de donnée et d'informations pour constituer le réseau social. Clicmobile a également mis à disposition son infrastructure pour héberger le serveur du projet (applicatif MoSoMuSo et système de contrôle et de versionnement).

Du côté du **LIP6** un état de l'art était en cours de rédaction présentant les différents travaux sur lesquels se basait le projet et les résultats futurs attendus. Par ailleurs, suite au stage de Guillaume Valadon chez Sony CSL, divers outils de tests étaient disponibles et j'ai pu orienter mon travail avec son aide.

### 1.4.1 État des connaissances

Les connaissances au sein du LIP6 se concentrent principalement autour des protocoles réseau et le laboratoire n'a pas réellement de spécialistes de programmation embarquée ou en système Symbian (fonctionnant sur la plateforme choisie). De ce fait, ma formation s'est faite au fur et à mesure à partir du guide Symbian OS [2] ou de notes d'applications fournies par le constructeur de la plateforme [3]. Les connaissances réseau m'ont été apportées grâce à la lecture de livres d'introduction au réseau de Guy Pujolle [?] et [4] Les connaissances plus spécifiques m'ont été apportées par mon maître de stage grâce à divers articles ([5] par exemple).

### 1.4.2 Missions

Afin de faciliter la compréhension des chapitres suivants je vais commencer par introduire les différentes missions et les choix qui ont déterminé le déroulement du stage. Les justifications techniques seront détaillées en 4.

#### Choix de la plateforme

Après une phase de tests de plusieurs plateformes et l'étude des besoins du projet le choix s'est porté sur les téléphones Nokia série N (Multimédia) et plus particulièrement le modèle N95. Clicmobile faisant partie du Forum Nokia Pro (collaboration et support aux entreprises utilisant les technologies Nokia), nous avons pu utiliser des prototypes de ce téléphone dès février 2007, soit 2 mois avant sa sortie officielle.

#### Réflexion et conception de protocoles de communication

Dans un premier temps il était nécessaire de développer un prototype fonctionnant simplement, il a donc été décidé d'implémenter une communication d'une part avec un serveur central puis, d'autre part avec d'autres utilisateurs. Ce dernier protocole se base sur File Transfert Protocol (FTP) pour l'échange de fichiers. Dans un deuxième temps une réflexion sur les protocoles ad hoc et sur le routage multi-sauts a été engagée afin de l'intégrer à court terme dans le prototype.

#### Définition du fonctionnement des prototypes

Dans un deuxième temps la définition exacte du fonctionnement du prototype a été réalisée. Cela afin de préparer des livrables pour diverses échéances, la première étant un événement organisé par Nokia, le *Nokia Summit Application* à Madrid en avril 2007 et le second étant un workshop tournant autour de la musique mobile à Amsterdam en mai 2007 (*Mobile Music Workshop* où nous avons publié un article [6]. La troisième venant avec la livraison d'un livrable au RIAM durant l'été 2007.

#### Développement du prototype

L'urgence était d'obtenir au plus vite une plateforme de test réseau et applicative afin de pouvoir valider le concept du projet. C'est cette phase qui a mis en avant les différents problèmes de la plateforme et du langage Python choisi pour le développement.

## Tests

Avant la première échéance représentée par le *Nokia Summit Application* le prototype n'avait jamais été testé en condition réelle à cause de problèmes liés au coûts et aux abonnements téléphoniques nécessaires aux tests. De ce fait un simulateur a été réalisé afin d'émuler les composants posant problèmes.

Ce simulateur a également été d'une grande aide lors des événement afin de présenter un scénario concret et explicite du concept aux visiteurs en intérieur et sans connectivité réseau.

## Présentation du travail réalisé

Le *Nokia Summit Application* à Madrid a été l'occasion de présenter le prototype et de recueillir les remarques et les avis du public de commerciaux et développeurs présents. Ensuite, au *Mobile Music Workshop* nous avons pu expérimenter le prototype avec un public radicalement différent (artistes, musiciens) et ainsi recueillir des remarques plus fonctionnelles et liés à l'utilisation.

### 1.4.3 Contexte et déroulement du stage

Après mon installation au LIP6 en février 2007 la première phase a été la formation aux réseau en général et plus particulièrement aux réseaux mobiles téléphoniques et WiFi. Dans un deuxième temps, le choix de la plateforme de développement s'est imposé comme une urgence afin de pouvoir travailler à l'implémentation.

Ensuite, l'installation des différents outils et kits de développement sur un environnement Windows a pu être réalisée. En effet le constructeur Nokia ne fournit qu'une version pour Windows de ses outils de développement pour téléphone mobile. La tentative d'utilisation du kit de développement sur Mac OS X s'est soldée par un échec et une perte de temps. En effet certains outils redéveloppés par la communauté open source se sont montrés complètement instables et incomplets.

Une fois le travail commencé j'ai pu apprécier la qualité des différentes documentations mises à disposition par Nokia pour les développeurs. D'autre part, nous avons pu compter sur la relation privilégiée avec Nokia que le Forum Pro nous offrait pour obtenir un support et de l'aide lorsque nous en avions besoin.

Le projet a été réorienté lors d'une réunion réunissant tous ses membres et c'est à ce moment que mon travail a vraiment démarré avec une ligne directrice, des jalons datés et trois objectifs, le *Nokia Summit Application*, le *Mobile Music Workshop* puis durant l'été 2007, un livrable pour présenter l'avancement du projet au **RIAM**.

Nokia rassemble chaque année tous les partenaires et utilisateurs de ses produits basés sur la plateforme Symbian/S60, les *Nokia Summit Application* et *S60 Summit*. Cette année, Clicmobile était invité à présenter son travail dans le cadre de sa participation au Forum Nokia Pro. De ce fait, nous avons présenté le prototype de **MoSoMuSo** et son concept.

Au retour de Madrid nous avons eu 2 semaines pour corriger les différents *bugs* qui ont été découverts avant de partir pour Amsterdam. Le *Mobile Music Workshop* est un événement organisé par le STEIM, Amsterdam (Center for research & development of instruments & tools for performers in the electronic performance arts) afin d'explorer le monde émergent de la musique mobile.

Nous avons présenté l'article écrit suite au développement du prototype [6] et nous avons expérimenté le prototype en conditions réelles lors d'une session "Hands On". Cela

a permis à des volontaires d'utiliser l'application, de la tester et ensuite de nous faire part de leurs remarques

Aujourd'hui le travail se poursuit sur une implémentation des protocoles de routage et sur la stabilisation du prototype pour effectuer une expérimentation à grande échelle sur le campus de Jussieu.

Un diagramme de Gantt explicite le déroulement de mon stage plus en détails sur la figure 1.1.

#### 1.4.4 Aspects organisationnels

Le projet est mené par différentes personnes au sein des différents partenaires, voici un aperçu des acteurs :

- Clicmobile, Samir Addamine, Alex Kummerman : Commercial
- Clicmobile, Lorys Pognon : Développeur
- Sony CSL, Atau Tanaka : Recherche sur la musique malléable
- LIP6, Naceur Malouch : Recherche sur les protocoles réseau
- LIP6, Bénédicte Le Grand : Recherche sur les mécanismes incitatifs en réseau
- LIP6, Guillaume Valadon : Recherche sur les modèles de mobilité

Mon poste d'ingénieur me donne le rôle de coordination du développement et de fourniture d'une plateforme utilisable pour que des recherches puissent être effectuées sur cette plateforme.

J'ai été encadré par Naceur Malouch mais j'ai été en relation avec l'ensemble des acteurs tout au long du projet. Selon les périodes j'ai travaillé avec différentes personnes ; Alex Kummerman pour la préparation du *Nokia Summit Application* puis Guillaume Valadon et Atau Tanaka pour la préparation du *Mobile Music Workshop*.

Des points de contrôle ont été effectués régulièrement afin de valider mon avancement, hebdomadairement avec mon maître de stage et un peu moins régulièrement avec les autres acteurs. Mais les outils mis en place pour la gestion du projet ont permis à l'ensemble des participants d'être tenu au courant grâce à une liste de diffusion pour les discussions propres au projet.

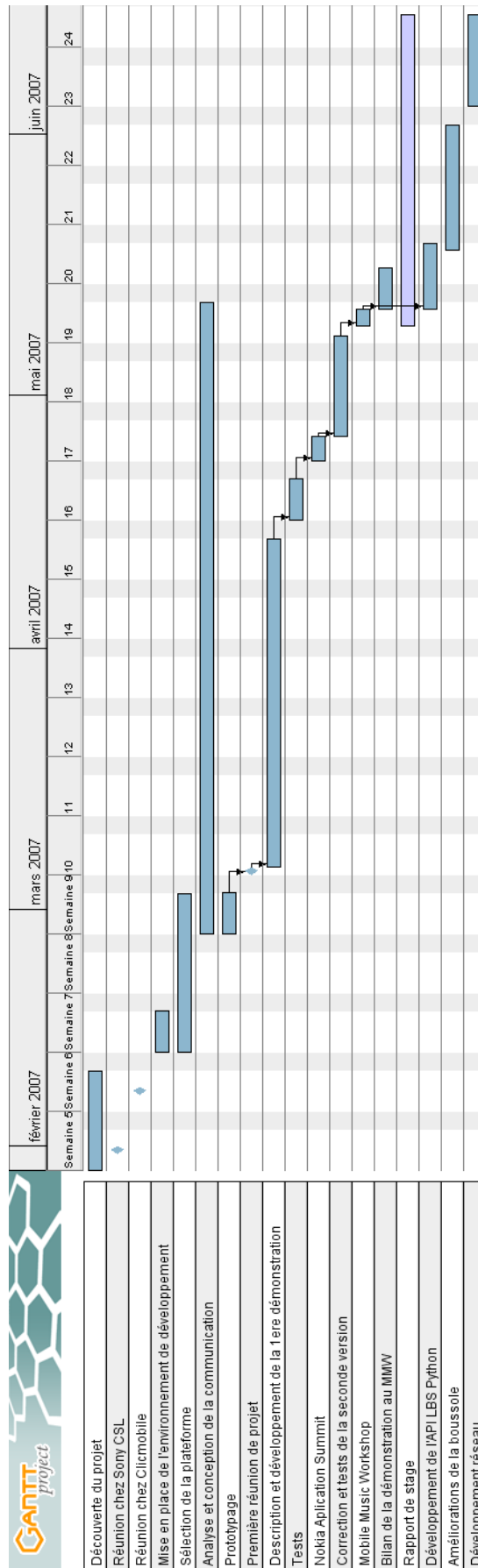


FIG. 1.1 – Déroulement du stage, diagramme de Gantt



## Chapitre 2

# Applications et réseaux sociaux

Le concept de MoSoMuSo, *Mobile Social Music Software* est innovant et se base sur de nombreux travaux dans les domaines des **MoSoSo**, *Mobile Social Software* mais aussi de la musique qui est, par essence, un média social et qui, aujourd'hui, est devenu mobile et qui introduit des contraintes de qualité exigeantes, aussi bien sur le transport que sur le rendu.

Nous commencerons par faire un aperçu des applications sociales typiques sur internet puis nous verrons les nouvelles modes dans les réseaux sociaux mobiles et l'introduction, dans ces applications, du concept de la spontanéité, chère au caractère humain. Dans une seconde partie une présentation de la musique mobile comme média social sera abordée et nous introduirons les concepts qui sont utilisés dans le projet.

### 2.1 Applications sociales

Commençons par définir ce qu'est un réseau social, son principe est de faire une carte des relations entre individus. L'analyse des réseaux sociaux est une technique utilisée en sociologie, anthropologie et en psychologie sociale qui aide à expliquer des phénomènes de la vie courante ou l'organisation des sociétés.

Depuis 2004, l'ère est aux réseaux sociaux virtuels et autres services de mise en relation ou de communautés. Parmi les nombreuses plateformes de réseaux sociaux qui se sont développées on peut compter la plateforme communautaires MySpace [7] qui permet de mettre à disposition son profil et partager ses contacts. Flickr [8] quand à lui est un exemple de réseau social appliqué à une certaine communauté, celle des photographes amateurs ou professionnels. En effet c'est une application *on-line* de gestion et de partage de photos. Mais le vrai principe de réseau social est plus clair dans MySpace ou encore JuiceCaster [9] qui permet de publier du contenu multimédia à partir de son mobile vers son profil et ainsi le partager. JuiceCaster souffre, par contre, d'un manque de membres actifs car un réseau intéressant est un réseau riche et populaire.

Les réseaux professionnels sont également en pleine expansion, inspirées des communautés d'amis celles-ci se basent sur les relations professionnelles et permettent d'élargir son carnet d'adresse grâce aux contacts de son réseau. LinkedIn [10] en est un exemple, ainsi que son concurrent européen Viadeo [11].

Les services de rencontre ne sont pas en reste car eux aussi ont adopté le créneau en élargissant leurs services. Passant de simples systèmes d'annonces, ils sont devenus, ces dernières années, de vraies plateformes communautaires. On peut donner l'exemple de Meetic [12] qui se met peu à peu à la mobilité avec Meetic Mobile, une version pour

terminaux mobiles de son site internet avec service d’alerte en direct. Mais c’est une simple adaptation du contenu à la plateforme de consultation, Meetic Mobile ne profite pas encore du potentiel de la plateforme sur lequel il fonctionne. Car les avantages sont bien réels, les informations de géolocalisation ou les statistiques d’utilisation de l’appareil sont en général disponibles et pourraient être utilisées.

C’est dans cette optique que Nokia Sensor [13] a été développé. il est le pionnier en terme d’application sociale mobile en permettant de contacter et de discuter via Bluetooth avec les gens présents à proximité. Elle n’a pas eu le succès escompté car le Bluetooth reste limité en terme de portée et Sensor n’était pas adossé à une communauté. De plus cette application nécessite d’être installée sur tous les mobiles et d’être lancée pour fonctionner. Ces contraintes ont été contournés par Mobiflirt [14] qui est, dans ce sens, le premier service de rencontre utilisant les avantages de la mobilité grâce au Bluetooth. L’application utilise la technique du **Blue Jacking** pour détecter des autres membres du réseau et pour les mettre en relation lorsqu’ils correspondent aux critères qu’ils ont enregistré dans leur profil. Mais en plus elle permet d’envoyer des messages aux personnes non présents dans le réseau (ou dont MobiFlirt ne tourne pas) et ainsi d’être moins limité en terme de public touché. De plus, après le premier contact, les personnes ont la possibilité de transmettre MobiFlirt et ainsi agrandir le réseau sur un principe de distribution virale (mais le destinataire doit accepter l’installation du logiciel). Même si le contacté n’accepte pas l’installation de MobiFlirt, les messages contenant une adresse **Wap**, ils pourront tout de même accéder au profil de la personne leur envoyant un message. Cette relative ouverture explique son succès par rapport à Nokia Sensor.

## 2.2 Applications sociales musicales

Les années 1970 ont vu arriver les « Walkman » de Sony qui ont rendu possible l’écoute de musique où que l’on soit mais depuis il y a eu peu de révolutions en terme d’usage. Le baladeur est toujours plus petit mais fonctionne toujours de la même manière, un support de stockage de musique, une commande de lecture et un casque audio pour l’écoute. Le partage se développe mais l’expérience musicale reste toujours personnelle et l’écoute commune passe encore par le prêt d’une oreillette du casque. Avec les nouvelles technologies de réseau sans fil il est maintenant possible d’effectuer un partage en direct et quelques applications se sont développées autour de ce concept.

Le concept commercial le plus avancé est le baladeur Zune [15] qui intègre le Wifi et propose un partage musical avec une gestion des droits de copie et un achat possible via la plateforme de Microsoft. Il ne propose pas l’écoute en streaming mais c’est le premier matériel dédié (baladeur numérique) intégrant l’idée de partage de musique sans fil. Il permet d’envoyer un morceau à un autre utilisateur de Zune et lui donner 3 jours pour l’écouter (ou 3 écoutes) avant qu’il ne soit invité à l’acheter ou à l’effacer.

Mais l’inconvénient ici est qu’il se limite à ce baladeur, c’est une solution matérielle alors que les solutions logicielles ont l’avantage d’être plus facilement déployable sur un grand nombre de périphériques. Ainsi, Zune introduit un usage lié au produit comme l’avait fait Apple avec l’iPod.

Le projet Push!Music [16] va plus loin en proposant un système d’agents automatiques qui “visitent” les baladeurs à proximité et récupèrent les morceaux de musique qui correspondent à certains critères basés sur les morceaux détenus, les lectures récentes et les styles préférés.

Mais le projet le plus proche de nos attentes est certainement tunA [17] qui est le pionnier d’une nouvelle génération de baladeurs permettant de partager l’écoute musicale.



tunA est un transmetteur radio **ad hoc** qui permet d'écouter ce qu'un autre utilisateur de tunA écoute. Il n'y a aucune copie de fichier musical, tout se passe en diffusion en direct ce qui supprime les problèmes de propriété intellectuelle. De plus, permettre à des gens d'écouter en même temps la même chose sur leur baladeur retrouve l'aspect social de la musique en évitant le prêt d'une oreillette.

## 2.3 Musique mobile malléable

Ce concept va plus loin que la musique mobile car il ajoute une dimension d'expérience communautaire.

Il se base au départ sur des formes de musique électronique créées à partir des mouvements des musiciens pour générer des sons et finalement un ensemble agréable à écouter et cohérent. Sensorband<sup>1</sup> en est un exemple et produit des concerts grâce aux mouvements sur scène des musiciens. Le plus gros problème de ce système réside dans sa complexité. Il n'est pas transportable alors que l'imagination de l'artiste ne s'arrête jamais. C'est pourquoi la mobilité est un axe de recherche pour permettre aux artistes de pouvoir créer en permanence.

La musique mobile malléable a été développée par Atau Tanaka travaillant chez Sony CSL [18] et consiste en un système collaboratif de création musicale sur dispositif mobile. Le système est composé d'un serveur de streaming utilisant la dynamique sociale et la mobilité comme paramètres pour re-mixer un fichier musical. La musique est ensuite renvoyée aux participants qui peuvent agir directement sur le son en manipulant leur dispositif ou en bougeant géographiquement. Le concept a été testé en grandeur nature lors de l'exposition *Intensive Science* organisée par Sony CSL à la Maison Rouge, Paris en octobre 2006. Le système est basé sur MAX/MSP (voir 2.3.1) qui effectue le traitement du son en temps réel. C'est cette application qui sert de base à **MoSoMuSo** et qui introduit les différentes problématiques de communication temps-réel.

### 2.3.1 Logiciel MAX/MSP

Le traitement temps réel de musique se base sur des algorithmes de traitement du signal très complexe et l'ensemble de ces fonctions sont intégrées dans un logiciel qui est la référence dans le domaine, nous allons donc décrire ici succinctement cette boîte à outils du musicien.

MAX/MSP est un logiciel destiné aux musiciens, il offre un environnement graphique de création musicale, audio et multimédia. Il est composé de MAX, MSP et Jitter.

MAX a été développé initialement à l'**IRCAM** pour permettre le contrôle temps réel d'instruments **MIDI**. Il était composé de l'environnement de programmation visuelle MAX et d'un composant d'exécution temps réel nommé *Faster Than Sound* (FTS).

MAX/MSP est aujourd'hui un produit commercial mais une version libre et gratuite est toujours développée à l'IRCAM, jMax.

MSP (MAX Signal Processing) remplace maintenant FTS et fournit les fonctionnalités de traitement audio temps réel allant du filtrage interactif à l'écriture sur disque.

Jitter est un composant qui offre la possibilité de manipuler des données matricielles optimisées pour la vidéo et le graphisme en 3 dimensions.

Un autre logiciel libre, Pure Data [19] est également dérivé de MAX/MSP et est développé au *Center for Research in Computing and the Arts* de l'Université de Californie.

---

<sup>1</sup><http://www.sensorband.com/>



## Chapitre 3

# Technologies réseau

De nombreuses technologies entrent en jeu dans ces problématiques pour permettre à ces applications de fonctionner comme, la puissance de calcul, l'accès au réseau Wifi ou le Bluetooth, ou encore des périphériques spécifiques tels que le récepteur GPS. Nous allons présenter ici les plus importantes et celles qui ont motivé nos choix techniques.

### 3.1 Techniques d'accès

La téléphonie mobile a beaucoup évolué ces dernières années pour offrir toujours plus de services et de débit, voici les quelques normes mises en jeu dans les communications mobiles et leurs caractéristiques.

#### 3.1.1 GSM

“Global System for Mobile Communications” ou **GSM** (historiquement “Groupe Spécial Mobile”) est une norme numérique pour la téléphonie mobile. Elle fut établie en 1982 par la Conférence des Administrations Européennes des Postes et Télécommunications (CEPT) et a été mise au point par l’Institut européen des normes de télécommunication (**ETSI**) et vient améliorer la première génération de réseaux téléphoniques mobiles tel que le “Nordic Mobile Telephone” ou “Radiocom 2000” qui étaient en service dans les années 1980. Cette norme est maintenant utilisée dans le monde entier avec des variantes, en Europe les bandes de fréquences des 900 MHz et 1800 MHz sont utilisées alors que sur le continent américain les fréquences sont 850 MHz et 1900 MHz.

Le réseau GSM est un réseau de télécommunication commuté basé sur un concept de maillage d’un territoire par des cellules qui couvrent une zone géographique pour fournir le service de téléphonie.

#### Architecture réseau générale

Le réseau GSM est constitué de quatre principaux éléments qui sont reliés via le réseau de l’opérateur au réseau téléphonique commuté public (**RTCP**) :

- La station mobile (MS) désignant le téléphone mobile ou tout autre appareil doté d’une interface GSM et disposant d’un accès au réseau via une carte **SIM**.
- La partie radio (**BSS**) composée d’un maillage de stations de base (**BTS**) et de leurs stations de contrôle (**BSC**). C’est à cet endroit que la voix est compressée et décompressée grâce à un transcodeur.

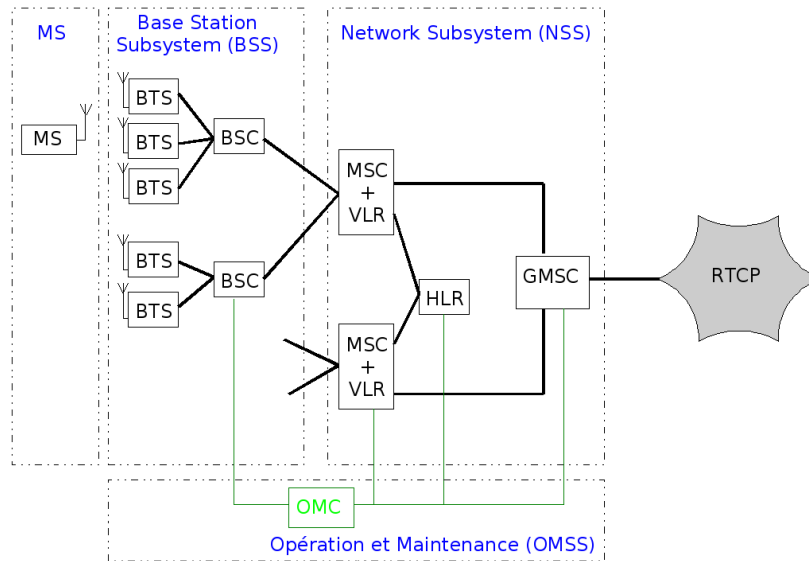


FIG. 3.1 – Structure simplifiée d'un réseau GSM

- La partie routage, appelée Network Subsystem (NSS) ou Core Network Subsystem, chargée d'acheminer la communication entre deux utilisateurs du réseau ou vers l'extérieur (réseau fixe par exemple) : elle est composée de Mobile Switching Centers (MSC) reliés entre eux qui effectuent le routage des communications. À chaque MSC est rattaché un Visitor Location Register (VLR) qui gère les informations des abonnés se trouvant dans la zone gérée par le MSC. Une base de données unique, le Home Location Register (HLR) gère la liste des abonnés de l'opérateur du réseau. Pour se connecter à d'autres réseaux extérieurs, il y a une passerelle dédiée, le commutateur réseau Gateway Mobile Switching Center (GMSC). L'AuC (Authentication Center) se charge de l'authentification des utilisateurs.
- La partie Opération et Maintenance, gérée par l'Operation and Maintenance Center (OMC), qui est chargé de mettre en place et de veiller au bon fonctionnement des différents éléments du réseau.

### Identification des utilisateurs, SIM

La SIM est l'application GSM qui permet de gérer l'identification des utilisateurs. Par abus de langage on appelle SIM la carte à puce qui contient cette application.

Un numéro d'identification unique (IMSI) est contenu dans la SIM et permet d'identifier un utilisateur sur le réseau GSM. C'est à partir de ce numéro que le réseau, via les MSC, autorise un utilisateur à se connecter au réseau et lui fournit un numéro d'identification temporaire (TIMSI, Temporary Mobile Station Identifier) sous lequel il sera identifié et joignable. Seul le HLR connaît la correspondance entre le numéro d'appel (MSISDN, Mobile Station ISDN Number) et l'IMSI/TIMSI et se charge du routage des appels entrant. Ceci afin de ne pas divulguer le numéro d'appel sur le réseau et ainsi garantir la confidentialité des appels.

A noter qu'il existe un autre numéro qui identifie de manière unique chaque terminal mobile, l'IMEI, mais ce numéro n'est utilisé que pour identifier le matériel par l'opérateur afin d'autoriser ou non l'accès à son réseau. C'est utile pour le blocage des téléphones mobiles volés.

### 3.1.2 GPRS

Le General Packet Radio Service (**GPRS**) est une norme pour la téléphonie mobile dérivée du GSM permettant un débit de données plus élevé, il représente la génération 2,5 (ou 2,5G). Le GPRS est une extension du protocole **GSM** : il ajoute par rapport à ce dernier la transmission par paquets. Cette méthode est plus adaptée à la transmission des données. En effet, les ressources ne sont allouées que lorsque des données sont échangées, contrairement au mode “circuit” en GSM où un circuit est établi et les ressources associées réservées pour toute la durée de la communication.

Le GPRS permet de fournir une connectivité IP constamment disponible à une station mobile (MS), mais les ressources radio sont allouées uniquement quand des données doivent être transférées, ce qui permet une économie de la ressource radio. Les utilisateurs ont donc un accès bon marché, et les opérateurs économisent la ressource radio. De plus, aucun délai de numérotation n'est nécessaire et, surtout, la communication peut être facturée aux données effectivement transférées et non au temps de communication. Une connexion GPRS est également possible en même temps qu'une session GSM voix, ce qui autorise l'utilisation de la téléphonie en même temps.

Avant le GPRS, l'accès à un réseau se faisait par commutation de circuits, c'est-à-dire que le canal radio était réservé en continu à la connexion. Cela ne nécessitait aucun matériel spécifique, le GPRS introduit de nouveaux équipements qui permettent d'établir une passerelle vers un réseau IP. Ces équipements, les **SGSN** (Serving GPRS Support Node) et **GGSN** prennent le relais dès le **BSC** prennent en charge les connexions sur un réseau spécifique qui dispose d'une passerelle vers le réseau internet.

La connexion entre le **MS** et le **SGSN** utilise différents protocoles de bas niveau (entre 1 et 3 dans le schéma **modèle OSI**) pour finalement utiliser un protocole qui met en place un tunnel IP, le **GTP**. À partir de ce moment une connexion TCP/IP peut avoir lieu entre le **MS** et un serveur distant en passant par le réseau de l'opérateur. Il est à noter qu'un mobile connecté en GPRS ne dispose pas d'une adresse IP publique et qu'il n'est pas joignable directement.

### 3.1.3 EDGE

L'**EDGE** (“Enhanced Data Rate for Gsm Evolution”) représente le successeur du GPRS (et ainsi la génération 2,75) car il repose sur une évolution du GSM et réutilise les principes du **GPRS** mais atteint des débits supérieurs grâce à l'utilisation (selon les conditions) du codage 8-**PSK** qui permet de coder 3 bits par symbole transmis alors que le **GSMK** utilisé en GPRS n'en transmet qu'un seul. À noter que l'**EGPRS** (amélioration du **GPRS**) supporte également cette modulation. De plus, l'**EDGE** permet d'utiliser plusieurs canaux en même temps ce qui permet encore d'augmenter le débit. L'inconvénient est que cette modulation est plus sensible aux perturbations et même en **EDGE**, lorsque l'environnement est trop bruyé, la modulation la plus robuste sera utilisée (et donc la moins rapide).

### 3.1.4 UMTS

L'**UMTS**, “Universal Mobile Telecommunications System” représente la 3ème génération de standards mobile. L'UMTS se base sur la technologie **W-CDMA** “Wideband Code Division Multiple Access”, le multiplexage par code large bande. L'UMTS permet des débit allant jusqu'à 2000 kbit/seconde mais cela varie beaucoup selon les conditions et ça peut

descendre à 144 kbit/seconde en zone rurale, ce qui reste toujours plus important que le GSM qui fonctionne à 9.6kbit/seconde.

L'UMTS a fait son apparition en France avec le lancement d'offres dites 3G et des services de visiophonie.

Il est défini par le consortium 3GPP et correspond à la norme WCDMA - 3GPP révision 99.

### 3.1.5 HSDPA

Le **HSDPA**, "High-Speed Downlink Packet Access", est une évolution logicielle de l'UMTS qui permet d'atteindre des débits de 1.8 Mbit/seconde, voire 3.6, 7.2 et 14.4 Mbit/seconde selon la version de la norme déployée dans le matériel. HSDPA correspond à la version 5 de la norme WCDMA par le 3GPP.

Les premières offres HSDPA (nommés 3G+) ont été lancées début 2007 en France. Une évolution est d'ores et déjà en développement, HSPA+ (ou HSPA Evolution), qui correspond à la norme 3GPP révision 7 et devrait atteindre des débits de 42 Mbit/s en réception et 11 Mbit/s en émission.

### 3.1.6 802.11

IEEE 802.11 est communément appelée Wifi ou WLAN et définit une norme pour les réseaux locaux sans fil. Son succès est principalement dû à son faible prix de revient et à l'absence de paiements de droits pour l'utilisation des bandes de fréquences des 2,4GHz et des 5GHz. l'inconvénient majeur est le partage de ces fréquences avec beaucoup d'autres applications (comme le micro-onde ou les téléphones sans fil d'intérieur) qui tendent à perturber les communications et ainsi abaisser les performances globales.

Elle se décline en différentes versions :

802.11a Utilise une bande de 300MHz dans la bande de fréquences de 5GHz et offre un débit de 54 Mbit/s sur une portée de quelques dizaines de mètres (30 Mbit/s effectifs).

802.11b Utilise une bande de 83MHz dans la bande de fréquences des 2.4GHz et offre un débit de 11 Mbit/s sur une portée de quelques centaines de mètres en terrain dégagé.

802.11g Compatible avec la norme 802.11g elle offre un débit de 54 Mbit/s sur la bande de fréquence des 2.4 GHz.

802.11n Norme qui utilisera les deux bandes de fréquence et qui promet des débits théoriques de 540 Mbit/s. Elle est en cours d'élaboration.

Le wifi n'est pas directement lié au monde de la téléphonie mais y est de plus en plus présent grâce à la généralisation de la voix sur IP (**VoIP**) et à son introduction sur les téléphones mobiles.

## 3.2 Méthodes de positionnement

La localisation reste une des problématiques les plus difficiles, la partie logique du réseau ne proposant pas de moyen d'être localisé il va nous falloir d'autres techniques pour disposer de cette information.

La récente loi américaine **E911** [20] sur les numéros d'urgence a fait jaillir les problèmes liés au positionnement. En effet, même les opérateurs ne disposent que d'une approximation de l'ordre de la cinquantaine de mètres en ville jusqu'à plus de trois kilomètres en campagne, ce qui est inacceptable pour une localisation d'urgence.

| Norme        | Débit maximum | Dénomination |
|--------------|---------------|--------------|
| <b>GSM</b>   | 9,6 kbit/s    | 2G           |
| <b>GPRS</b>  | 115 kbit/s    | 2.5G         |
| <b>EDGE</b>  | 384 kbit/s    | 2.75G        |
| <b>UMTS</b>  | 2 Mbit/s      | 3G           |
| <b>HSDPA</b> | 14.4 Mbit/s   | 3G+          |
| 802.11a      | 54 Mbit/s     |              |
| 802.11b      | 11 Mbit/s     |              |
| 802.11g      | 54 Mbit/s     |              |
| 802.11n      | 540 Mbit/s    |              |

TAB. 3.1 – Récapitulatif des normes et débits

Plusieurs techniques sont utilisées aujourd’hui pour ce genre de services que nous allons voir maintenant.

### 3.2.1 Localisation basée sur le réseau

Orange commercialise un système de géolocalisation basée sur le réseau [21], le système utilise les cellules GSM des stations de base pour trianguler la position du mobile. La précision annoncée est de l’ordre de 100 à 300 mètres en ville et de 300 mètres à 3 kilomètres en zone rurale. La donnée fournie est une position ainsi que la précision de la mesure (rayon autour de la position dans laquelle se trouve le mobile). La localisation est faite de différentes manières :

- Angle d’arrivée (**AOA**, cela requiert au moins deux stations de base et utilise l’angle par lequel le signal arrive à la station pour déterminer une triangulation. Habituellement les stations de base sont équipées de plusieurs antennes GSM directionnelles ce qui permet de disposer de cet angle.
- Différence temporelle (**TDOA**) qui est une technique de positionnement hyperbolique, il déduit la distance à chaque station de base pour déterminer une triangulation. C’est la méthode qui est aujourd’hui la plus utilisée.
- Signature de position qui se base sur les cellules GSM visibles à un endroit donné, cette technique est beaucoup utilisée par des services communautaires qui visent à répertorier les cellules et leur localisation pour fournir des services libres et peu onéreux de positionnement (comme [22]).

D’autres solutions hybrides existent et combinent ces techniques ou utilisent un récepteur GPS en plus.

### 3.2.2 Basée sur le matériel : le GPS

Le **GPS** ou système de positionnement mondial est le principal système de positionnement par satellite mondial actuel et actuellement le seul à être entièrement opérationnel. Ce système mis en place par le Département de la Défense des États-Unis peut permettre à une personne de connaître la position d’un objet sur la surface de la terre dès l’instant que celui-ci est équipé du matériel nécessaire au fonctionnement du système.

Le système GPS comprend au moins 24 satellites artificiels placés sur une orbite à 20200 km d’altitude. Ces satellites émettent en permanence un signal complexe (code pseudo-aléatoire) daté précisément grâce à leur horloge atomique, ainsi que des éphémérides permettant le calcul de leurs coordonnées prédites.

Ainsi un récepteur GPS qui capte les signaux d'au moins quatre satellites peut, en mesurant les écarts relatifs des horloges, connaître sa distance par rapport aux satellites et, par trilatération, situer précisément en trois dimensions n'importe quel point placé en dessous des satellites GPS (avec une précision de 15 à 100 mètres pour le système standard). Le GPS est ainsi utilisé pour localiser des véhicules roulants, des navires, des avions, des missiles et même des satellites évoluant en orbite basse.

un récepteur GPS a besoin de capter le signal d'au moins 4 satellites afin de se positionner précisément. Avec 3 satellites le positionnement en altitude est incorrect mais les longitudes et latitudes restent précis.

Les récepteurs GPS se sont démocratisés ces dernières années mais souffrent encore de multiples problèmes liés à leur forte consommation énergétique d'une part. Et d'autre part ils ont tendance à être relativement lent lors d'un démarrage à froid. En effet, après une longue période d'inactivité les éphémérides des satellites doivent être téléchargées ce qui peut prendre quelques dizaines de secondes. Ensuite la précision de la mesure dépend fortement de l'environnement, en ville des problèmes de réflexion sur les immeubles rendent bien souvent les récepteurs GPS imprécis voir complètement aveugles.

### 3.2.3 Basée sur les points d'accès WiFi

Une technique que le projet MoSoMuSo souhaite développer est le positionnement suivant des points d'accès WiFi visibles. Cette technique est proche de celle utilisée par les services comme [22], elle consiste à stocker les points d'accès vu par les utilisateurs et leur localisation afin d'utiliser ces informations pour localiser d'autres utilisateurs n'ayant pas de moyens de se positionner.

## 3.3 Systèmes d'exploitation mobile

Les périphériques mobiles bénéficient de systèmes particuliers intégrant les contraintes de temps réel et d'utilisation mémoire pour un fonctionnement qui peut aller jusqu'à plusieurs mois sans redémarrage.

### 3.3.1 Windows Mobile

Windows Mobile est le dérivé du célèbre système d'exploitation pour les plateformes mobiles. Il prend en charge un nombre important de périphériques et offre un panel important d'applications. Il permet entre autres de faire fonctionner *MSN Messenger* et *Microsoft Office*. Windows Mobile est un concurrent direct de *Blackberry* depuis l'introduction de technologies qui lui permettent de recevoir des courriers électroniques en temps réel. En 2005 il occupait 50% du marché des "smartphones".

### 3.3.2 Symbian OS

est un système d'exploitation pour plateformes mobiles dérivé du système Epos développé Psion pour ses appareils mobiles dans les années 1980. Symbian fonctionne exclusivement sur processeurs **ARM**.

Symbian Inc. développe aujourd'hui le système de base qu'il propose sous license aux différents constructeurs. À la charge ensuite des constructeurs de bâtir par dessus une interface graphique et de modifier le système selon leurs besoins.



La plateforme la plus commune est actuellement Series60 <http://www.s60.com> qu'utilise Nokia, LG Electronics, Lenovo, Samsung et Siemens. L'autre grande plateforme est UIQ <http://www.uiq.com> utilisée principalement par Sony Ericsson et Motorola.

Symbian a l'avantage d'être un système léger et adapté à la mobilité conçu autour d'un micro-noyau multi-tâches préemptif. Sa philosophie se base sur l'économie de mémoire et l'"événementiel" qui permet de couper le processeur lorsqu'il n'est pas utilisé. Il supporte un vaste panel de périphériques intégrés aux appareils mobiles, appareils photo, interface GSM, Bluetooth, WiFi, GPS...

### 3.3.3 GNU/Linux

Le système d'exploitation libre Linux est également de plus en plus utilisé dans les appareils mobiles. Il a l'avantage d'offrir aux programmeurs un environnement connu et standard. Des projets comme OpenMoKo (<http://www.openmoko.com/>) ou Greenphone de la société Trolltech (<http://www.trolltech.no>) utilisent ce système.

## 3.4 Protocoles réseau

MoSoMuSo comporte une très forte composante réseau sans-fil pour lui permettre d'établir la communication au sein d'une communauté et ainsi pouvoir partager l'expérience musicale. On peut voir cette communication intra-communautaire comme un réseau **ad hoc** mobile et plus particulièrement un réseau auto-organisable sans fil ou **SON**, Self Organizing Network. Après un aperçu des SON nous nous intéresserons à des protocoles de plus haut niveau qui sont utilisés dans le transport temps réel de contenu multimédia.

### 3.4.1 Réseaux auto-organisables

Les SON sont des réseaux sans infrastructure fixe qui sont composés de noeuds autonomes. Nous allons considérer des noeuds sans connexion filaire mais équipés d'une interface WiFi. Ainsi, chaque noeud ne peut communiquer qu'avec ses voisins qui sont à portée. Les voisins se chargent de relayer les messages pour qu'ils atteignent leur destination. de plus les noeuds sont mobiles ce qui génère un changement permanent dans la topologie du réseau formé.

Les SON se gèrent automatiquement c'est à dire que le routage des paquets dans le réseau est calculé dynamiquement et change selon les mouvements des noeuds. La topologie du réseau varie donc dans le temps.

La plupart des travaux existants partent du principe que la mobilité est subie et agissent donc sur les paramètres sous contrôle, c'est à dire la puissance d'émission de chaque noeud pour obtenir une topologie réseau définie. Dans notre cas la problématique est plus proche du concept développé par le projet Ambience [5] qui travaille sur des réseaux ad hoc en utilisant des routeurs mobiles qui se déplacent pour garantir une connectivité et la conserver. Ceci est fait grâce à un protocole de routage, nous allons en voir un maintenant, **OLSR**.

### 3.4.2 Optimized Link State Protocol

**OLSR** est un protocole proactif de routage utilisé dans les réseaux mesh ou ad-hoc multisautes. C'est un protocole d'état de liens qui permet de collecter des informations à propos des noeuds et réseaux accessibles et qui se charge de transmettre les messages en

utilisant chaque noeud comme un relais. Il n'y a pas de contraintes de qualité de service dans ce protocole mais une extension ajoute cela QOLSR [5].

## 3.5 Protocoles réseau niveau transport

La qualité de service n'est pas forcément faite au niveau du protocole de niveau transport, nombre de protocoles intègrent également des contraintes de qualité de service et d'optimisation afin d'améliorer leur fonctionnement. Nous allons voir maintenant un protocole (**RTSP**) qui nous intéresse fortement dans le sens où il permet le transport de flux multimédia en temps réel et qu'il est optimisé pour fonctionner sur des réseaux IP limités comme peuvent l'être ceux des opérateurs de téléphonie mobile. Nous verrons ensuite une alternative, le protocole **XMPP** qui intègre également des mécanismes de présence et de messagerie.

### 3.5.1 RTP/RTCP

RTP [23] qui remplace [24], "real-time transport protocol" fournit des fonctions de transport de données soumises à des contraintes de temps réel sur des réseaux unicast ou multicast. Par contre il ne garantit pas une qualité de service. Au transport de la donnée est ajouté un protocole de contrôle, RTCP "real-time transport protocol", destiné au suivi et à la transmission d'informations sur la session (participants, qualité de la distribution des paquets RTP).

RTP fonctionne sur UDP et transporte des informations sur le codage des données transportées, le numéro de séquence de chaque paquet, l'identification de la source et l'horodatage pour la synchronisation éventuelle avec un autre flux (pour le couplage audio/video par exemple).

### 3.5.2 RTSP

**RTSP** [25] est un protocole applicatif destiné à la transmission de flux multimédia avec une contrainte de temps réel. Il définit un système pour la transmission à la demande de contenu multimédia aussi bien direct que stocké sur le serveur qui se charge de la diffusion. Il fonctionne sur RTP ou plus généralement sur HTTP.

### 3.5.3 Jabber/Extensible Messaging and Presence Protocol

**XMPP**, "Extensible Messaging and Presence Protocol" [26] [27] définit un protocole de présence, messagerie et communication temps réel décentralisé basé sur XML. Le protocole à l'avantage d'être simple, ouvert et simplement compréhensible pour être facilement étendu. L'intérêt pour le projet est que ce protocole intègre depuis peu le support du transport audio. Ce qui en fait un protocole très complet aussi bien pour la présence et l'échange de messages que pour l'échange audio. Son principal défaut est qu'il se base sur XML et qu'il est très verbeux, il implique donc un très fort trafic réseau.

## Chapitre 4

# Analyse et conception

Lors de mon stage j'ai été amené à concevoir et à développer l'application de test suivant les besoins du projet. Cette application est finalement destinée à servir de support à la musique malléable. Mais avant de pouvoir l'y intégrer, l'application va servir de plateforme d'évaluation du matériel et du concept. Mais elle servira surtout pour effectuer une expérimentation grandeur réelle. Le but de l'expérimentation étant, d'une part, de collecter des informations sur les usages du téléphone et sur les modèles de mobilité des personnes et, d'autre part de mesurer les performances des protocoles conçus pour la connectivité et la qualité de service (QoS). Ces objectifs doivent être pris en compte lors de la conception pour proposer dès le départ une plateforme la plus complète possible.

### 4.1 Fonctionnement général

La première étape a été la réalisation d'un cahier des charges (informel), celui-ci devait inclure l'ensemble des besoins des différents partenaires afin de définir au mieux le fonctionnement de l'application et les technologies requises.

Lors de la première réunion de projet avec l'ensemble des partenaires, le fonctionnement général du prototype a été discuté dans l'optique du *Nokia Summit Application* et du *Mobile Music Workshop*. La figure 4.1 démontre le fonctionnement général de l'application avec un serveur central, une interface 3G donnant accès au serveur **MoSoMuSo** à travers l'internet et une interface WiFi utilisée pour les interactions entre utilisateurs. Le GPS servant pour la localisation.

### 4.2 Approches

Une des principale problématique introduite par le projet est le transfert de données entre mobiles, en temps réel ou non. Ceci nous a conduit à réfléchir à plusieurs approches que nous détaillerons ci-dessous.

Sans contraintes de temps réel, la solution des réseaux tolérants aux délais ("Delay Tolerant Network" - DTN) est envisageable. Cette solution accepte un délai dans la diffusion d'un message. Dans notre cas cela consisterais à "demander" aux fichiers de faire le déplacement. Cette approche est utilisée par Push!Music [16].

Vient ensuite la solution de l'utilisation de toutes les collectivités disponibles (Bluetooth, WiFi en mode infrastructure, 3G) afin de conserver une connexion sans se soucier du coût. Cette solution est utilisée par tunA [17] et par Juice Caster [9]. L'inconvénient

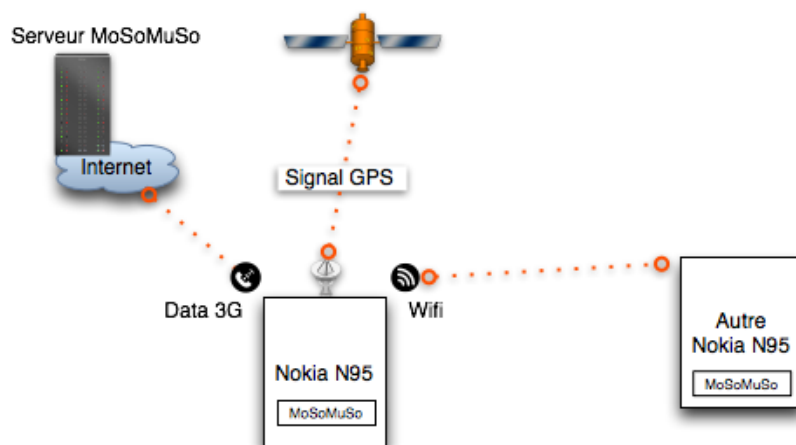


FIG. 4.1 – Fonctionnement général de l'application

est que cette dernière application suppose toujours avoir une connectivité réseau et dans le pire des cas utilise la couverture globale de la 3G qui est coûteuse.

La troisième solution est d'inciter les utilisateurs à bouger afin de trouver ou de conserver une connectivité réseau.

C'est cette troisième solution que nous avons exploré. Au premier abord cette approche semble irréaliste mais les utilisateurs qui sont attirés par quelque chose sont prêts à bouger, c'est en partant de cette constatation que nous avons développé un outil pour les y aider et les y inciter, la boussole.

L'hypothèse de départ de notre système est qu'il n'y a pas d'interaction possible dans la zone actuelle et que l'utilisateur va devoir chercher une zone plus intéressante. C'est une approche complètement différente des autres applications musicales mobiles comme tunA ou Push!Music qui partent du principe qu'il y a toujours une interaction possible et qui ne proposent pas de solution pour en trouver.

Par conséquent, le premier prototype a été défini comme étant une application devant pouvoir servir à l'utilisateur pour parcourir la liste de ses amis, voir leur état et, s'ils sont connectés et à proximité géographique, pouvoir les rejoindre pour interagir avec eux par un échange de fichiers.

Cette problématique a introduit plusieurs choix systèmes et réseaux décisifs pour l'application. Partant du principe que le téléphone mobile est aujourd'hui un formidable outil qui permet les interactions spontanées entre les personnes nous avons décidé de faire fonctionner le prototype sur ce type d'appareils. De plus, les mobiles actuels ne sont plus simplement utilisés pour converser, mais ils servent aussi à échanger des messages texte, prendre et envoyer des photos et des vidéos et écouter de la musique. Et le téléphone mobile est également aujourd'hui le gardien privilégié du carnet d'adresse, ce qui en fait un formidable outil social.

Le besoin de guidage physique (de "navigation") nous a conduit à intégrer une boussole afin de guider les utilisateurs vers leurs amis ou vers des points d'intérêt comme le fait un système de navigation de voiture.

Le partage des informations afin de pouvoir transmettre et mettre à disposition des autres utilisateurs les coordonnées géographiques a introduit le besoin d'un serveur central. Le serveur **MoSoMuSo** centralise donc les profils des membres ainsi que leurs liste de contact afin de pouvoir partager ces informations avec leurs amis mais également avec les

amis de leurs amis. De plus, la puissance du serveur par rapport au téléphone mobile nous conduit à effectuer le maximum de calculs sur le serveur avant de rapatrier les données sur le client. Cela permet d'accélérer légèrement le fonctionnement de l'application et c'est donc moins coûteux en consommation d'énergie (batterie).

### 4.3 Réseau social

C'est le coeur de l'application qui sert à former les communautés et à permettre les interactions entre les personnes. Le réseau social doit contenir l'ensemble des informations personnelles des utilisateurs et permettre le partage avec les personnes autorisées des informations adéquates.

Données requises pour l'identification d'un utilisateur :

- Nom, Prénom
- **MSISDN** (numéro de téléphone)
- Mot de passe

Données requises pour la partie réseau social :

- Position (latitude, longitude)
- Date de la dernière connexion à l'application
- Liste d'amis
- Éventuellement type de contenu multimédia proposé et détenu

Pour le stockage des données plusieurs solutions étaient possibles, la première, distribuée impliquait de mettre en place une architecture complexe basée sur un système décentralisé de stockage répartis. Cette solution impliquerait une charge de calcul et de transfert réseau trop important et donc, peut être, trop coûteuse. La seconde, validée par Clicmobile, était d'utiliser un serveur central pour stocker et centraliser les profils afin d'assurer une disponibilité permanente de ces informations en tout lieu.

Dans le prototype le serveur permettait également de stocker la liste des genres musicaux disponibles sur le téléphone ce qui permettait d'effectuer des recherches suivant le genre de musique demandée par un utilisateur et de trouver ceux, à proximité, qui s'intéressent au mêmes genres. Les titres exacts n'étant disponibles qu'après contact avec l'utilisateur.

Clicmobile ayant une forte expertise en réseaux sociaux sera capable de fournir l'architecture de l'application de gestion du réseau social. Mais pour la phase de prototypage il a été décidé de développer une plateforme basée sur le framework (ou cadre d'application) Django<sup>1</sup> afin d'obtenir rapidement l'application gérant le réseau social.

### 4.4 Boussole et positionnement

Le réseau social fonctionnant avec la position des utilisateurs, un système précis de localisation est requis. Le système de positionnement devra être capable de guider un utilisateur à proximité d'un autre afin de pouvoir effectuer une connexion ad hoc WiFi. L'erreur possible est donc de l'ordre d'une centaine de mètres afin de rester dans la zone de portée WiFi. Par contre la navigation doit être plus précise afin de guider l'utilisateur au mieux car une erreur d'un degré au départ peut donner plusieurs kilomètres à l'arrivée (même si dans notre cas la boussole se met à jour en permanence).

Les différents systèmes de positionnement étudiés en 3.2 nous ont conduit à choisir le **GPS** dans un premier temps et l'intégration de la méthode basée sur les points d'accès

---

<sup>1</sup><http://www.djangoproject.com>

WiFi le sera plus tard. Celle basée sur le réseau (voir 3.2.1) réseau GSM entraînant un coût pour l'utilisateur il a été décidé de ne pas la supporter.

La boussole a été pensée comme un instrument de navigation, aussi simple que possible afin de ne pas induire les utilisateurs en erreur. Le principe est d'indiquer par une flèche la direction vers laquelle l'utilisateur doit se diriger et mettre à jour la direction en fonction des déplacements. La figure 4.2 présente la première esquisse que j'ai faite en collaboration avec un designer de chez Climobile.



FIG. 4.2 – Esquisse de la boussole

La boussole est une composante du protocole réseau que nous allons créer car elle permet de former le réseau et d'agir sur sa topologie. De ce fait il est important de soigner son aspect et son fonctionnement afin de la rendre la plus efficace possible.

## 4.5 Connectivité réseau

Le téléphone mobile choisi met à notre disposition plusieurs types de connectivités réseau, d'une part le réseau téléphonique par l'intermédiaire du 3G offrant également une connexion données relativement rapide mais étant très limitée (voir 3.1.4). D'autre part une interface WiFi mais souffrant d'une portée réduite à une centaine de mètres et d'une consommation électrique importante. L'avantage de la seconde réside dans la gratuité des transferts. En effet la connexion donnée 3G est facturée au transfert et cela peut rapidement coûter très cher (voir 4.8.4 pour une estimation).

D'après ces constatations l'idée de proposer également un service de localisation de bornes WiFi ouvertes et proposant un accès internet a germé. Cela apportait deux avantages, d'une part commercial, d'après les retours que j'ai reçus, l'immense majorité des utilisateurs de téléphones mobiles proposant des services internet sont freinés dans leur utilisation à cause du coût. D'autre part, inciter les utilisateurs de MoSoMuSo à partager

avec leur réseau social permet également d'enrichir la base de points d'accès connue et géolocalisés ce qui permet ensuite d'utiliser ces informations pour positionner des utilisateurs qui ne seraient pas équipés d'un récepteur GPS sur le principe décrit en 3.2.3.

## 4.6 Interface utilisateur

L'interface utilisateur est un paramètre important dans une application. En effet c'est sur ce point que les utilisateurs apprécient une application. Nous avons décidé, suite à des problèmes purement implémentatoires, d'utiliser un ensemble d'images pour former l'interface graphique ce qui nous a permis de créer un produit d'une très grande qualité visuelle. L'OpenGL était une autre solution envisagée mais pour le prototypage elle a été laissée de côté afin de privilégier la rapidité.

## 4.7 Scénario d'utilisation

Le fonctionnement type que nous avons souhaité mettre en place repose sur le scénario suivant :

- Recherche de contenu et disponibilité à proximité
- Navigation avec la boussole en utilisant l'interface réseau 3G
- Rapprochement du but sélectionné
- Démarrage de l'interface réseau locale sans fil
- Précision des critères de recherche sur le contenu
- Échange de contenu

La recherche de contenu est intentionnellement très vague afin d'obliger le contact entre utilisateurs pour voir précisément les titres disponibles.

### 4.7.1 Fonctionnement de l'application

Après avoir défini l'objectif et les aspects d'utilisation de l'application, vient le besoin de créer sa structure fonctionnelle. Nous allons détailler grâce aux figures suivantes le fonctionnement de l'application. Pour commencer, la modélisation simplifiée, visible sur la figure 4.4 met en valeur l'organisation des différents modules qui sont développés sous forme de classes séparées. Le motif de conception utilisé dans l'ensemble de l'application est Modèle-Vue-Contrôleur car c'est celui qui est utilisé principalement dans le développement d'applications sur le système Symbian et nous avons voulu garder une cohérence globale.

Le contrôleur, *Core* se charge d'instancier et de gérer tous les autres objets. Les Vues sont composées de *CompassUI* qui s'occupe de l'affichage de la boussole ; *FileExchangeUI* qui est l'interface à l'échange de fichier ; *AddGoalUI* qui permet de sélectionner un but pour la boussole et enfin *GeotagUI* qui permet de géotagger un lieu et de l'ajouter au réseau social en l'annotant.

Les modèles sont représentés par différentes classes, d'une part *IGPS* qui est une interface aux différentes implémentations d'interrogation GPS et d'autre part *MoSoMuSoNetwork* qui est l'interface au réseau social. Cet objet gère d'une part l'interrogation du serveur central et d'autre part les connexions entrantes d'autres utilisateurs qui souhaitent échanger de la musique par l'intermédiaire de *Fe\_Server*. Le client d'échange de musique (*Fe\_Client*) est quand à lui directement inclus dans *FileExchangeUI*. Le fonctionnement de *Core* est défini par la figure 4.3.

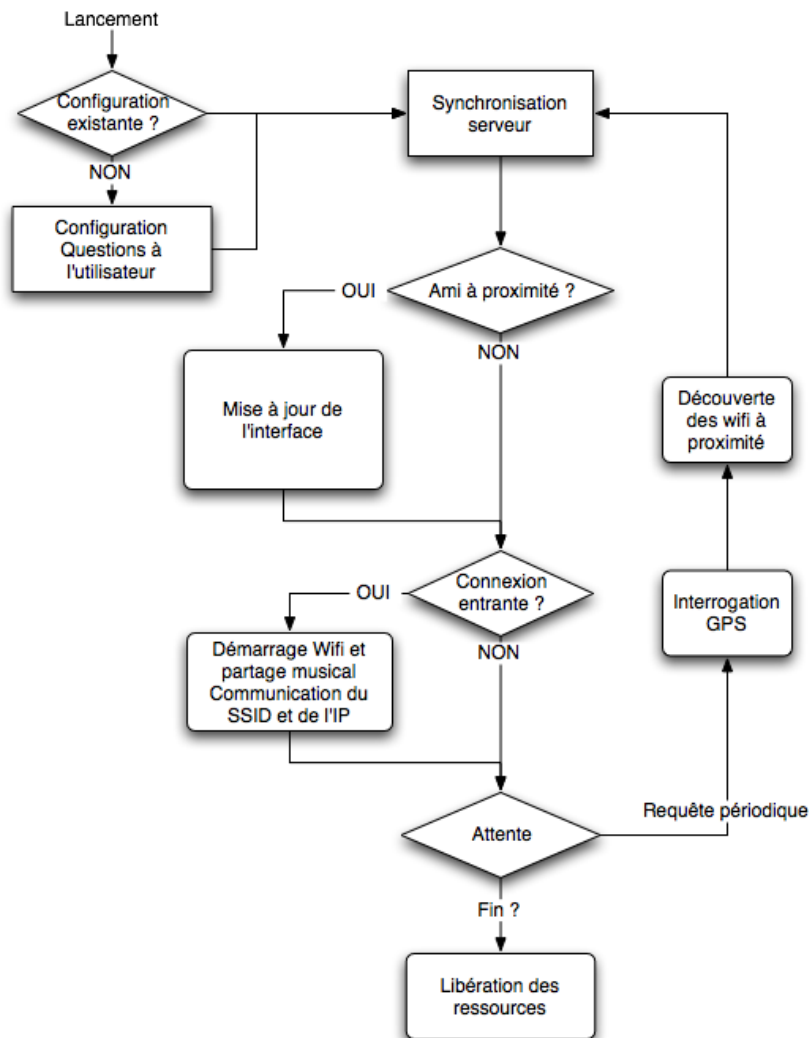


FIG. 4.3 – Fonctionnement logique de l'application

Dans la suite nous allons nous intéresser plus particulièrement au réseau social et à son implémentation.

#### 4.7.2 Interactions réseau

Les échanges réseaux ont été défini de manière à rester compréhensible facilement par un humain durant la phase de prototypage. Ainsi le protocole de présence Jabber (voir 3.5.3) n'a pas été utilisé. La communication entre le serveur et l'application **MoSo-MuSo** est faite par méthodes HTTP POST, la technique utilisée par les navigateurs pour soumettre des formulaires. L'ensemble des données transitant en clair sur le réseau.

### 4.8 Rafraîchissement serveur

Pour que le réseau social garde tout son intérêt, il est important que les informations disponibles ne soient pas périmées. Elles sont donc mises à jour régulièrement. Cela à un



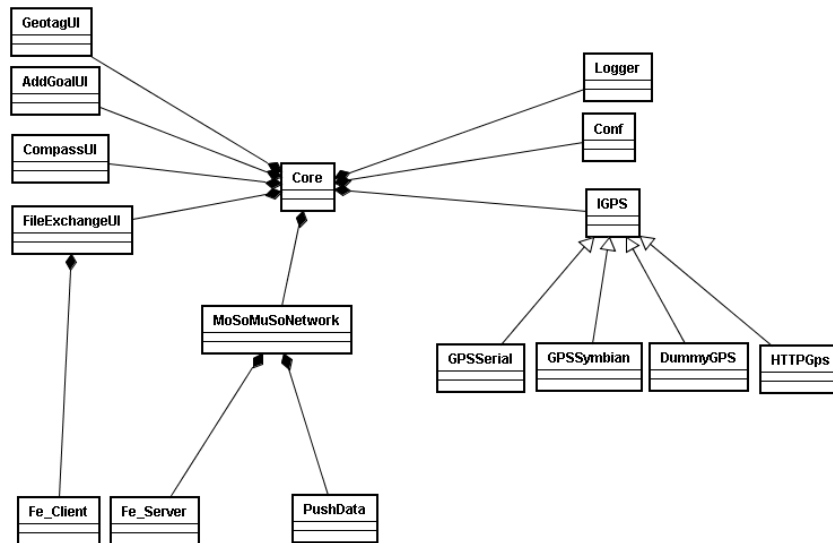


FIG. 4.4 – Modélisation de l’application (prototype)

coût induit par le trafic que l’actualisation des données génère sur le réseau de l’opérateur mobile.

#### 4.8.1 Actualisation de la localisation

La mise à disposition régulière de sa propre localisation est la condition pour que le système fonctionne. Il y a deux différentes situations, stationnaire et en mouvement qui n’impliquent pas la même fréquence de rafraîchissement. Dans le premier cas, l’actualisation des données n’est nécessaire que pour donner “signe de vie” et indiquer que l’utilisateur est toujours actif et disponible à une position donnée. Dans le second, l’actualisation de la position permet de tenir au courant les autres utilisateurs de sa dernière position et indiquer au système le mouvement.

On peut se demander si l’actualisation de la position est pertinente en mouvement car il sera très difficile pour un autre utilisateur de suivre à cause du délai introduit par la mise à jour de la position et sa récupération au niveau du suiveur, mais on peut imaginer qu’en milieu urbain le mouvement n’est pas constant mais qu’il se fait à très faible vitesse et ponctué d’arrêts. Réciproquement, un utilisateur ayant une vitesse trop importante peut être considéré comme inactif car il ne pourra que très difficilement être rejoint. C’est pourquoi le concept de mouvement relatif doit être introduit, deux utilisateurs se déplaçant à la même vitesse dans la même direction ne pourront jamais se rejoindre, il faut donc signaler ce type de situation. De même, un utilisateur prenant tout à coup le bus faussera le système et se rendra inaccessible aux utilisateurs de sa zone géographique. Mais par contre il se rendra sûrement disponible pour d’autres utilisateurs.

#### 4.8.2 Actualisation de la liste d’amis

La liste d’amis doit, elle aussi, rester à jour afin de permettre à l’utilisateur de connaître l’état de tous ses contacts et de lui permettre de naviguer vers eux. Un défaut de mise à jour pourrait conduire l’utilisateur à choisir une cible qui a changé de localisation ou qui a désactivé son **MoSoMuSo**.

De par la nature des informations échangées, la mise à jour de la liste d’amis est

une opération coûteuse en terme de trafic car elle se charge de récupérer la localisation géographique des amis ainsi que des renseignements sur leur état. De plus elle nécessite de récupérer l'ensemble des informations sur les amis d'amis (nom, numéro, localisation) car ils n'appartiennent pas nécessairement au carnet d'adresse de l'utilisateur.

La première optimisation est l'actualisation incrémentale, c'est à dire que seuls les changements depuis la dernière actualisation sont transmis.

La seconde est la limitation du nombre d'actualisation au strict minimum. Il est bien sûr impossible de déterminer quand un ami va se connecter ou se déconnecter mais il est possible de choisir un interval de temps optimal selon le comportement du réseau. C'est à dire si les amis ont tendance à bouger vite la fréquence de rafraîchissement sera plus rapide que si le réseau est assez statique.

### 4.8.3 Obtention de la position

La navigation à l'aide de la boussole introduit de nombreux problèmes. Le GPS est très pratique pour se repérer mais dès qu'il s'agit de faire de la navigation sans cartographie les choses se compliquent. Du fait de la vitesse très basse de déplacement de nos utilisateurs il est difficile de connaître précisément le cap (sens de déplacement). De plus, les GPS embarqués actuellement dans les téléphones mobiles ont très souvent une imprécision qui approche les 80 mètres, il est donc impossible de ne se baser que sur les dernières mesures pour évaluer le sens de déplacement.

D'un point de vue utilisateur, il est très déstabilisant de se trouver face à une boussole dont l'aiguille oscille de plus de quelques degrés. Il est donc important que la boussole soit précise ou, au moins, donne une bonne approximation de la direction de la cible sans osciller. Une boussole ne donnant qu'un cap approximatif au départ et s'ajustant au fur et à mesure du déplacement reste l'idéal.

Pour cela, un filtre de Kalman [28] va devoir être introduit afin de corriger toutes les incertitudes. Le filtre de Kalman, du nom de son créateur Rudolf E. Kalman, mathématicien américain, a été présenté en 1960. Il s'agit d'un estimateur récursif optimal de processus aléatoire. Il permet l'estimation de variables d'état d'un système (position/vitesse par exemple) en utilisant un modèle de comportement à priori du système et des mesures réelles. Son appellation "filtre" vient du fait qu'il permet, entre autres, d'estimer un signal bruité en séparant le signal de son bruit. Ce filtre est utilisé en trajectographie pour estimer une trajectoire, en navigation pour localiser un engin, ou encore en balistique pour estimer des conditions initiales inconnues. Il est également employé lorsque l'on veut coupler des informations venant de sources différentes afin de retrouver, par exemple, une trajectoire. Ce dernier cas serait l'utilisation idéale dans notre problématique mais nous n'avons malheureusement pas accès à un matériel dédié comme un accéléromètre.

Le filtre de Kalman sera donc utilisé pour filtrer les informations de positionnement données par le GPS et ainsi éviter de prendre en compte des données complètement aberrantes.

### 4.8.4 Coût en terme de trafic

Aujourd'hui les opérateurs mobile mettent à disposition différentes formules et forfaits pour permettre à leurs abonnés d'utiliser l'internet où qu'ils soient. Nous donneront ici une estimation du coût de fonctionnement de notre application celle ci n'étant pas encore optimisée grâce à la compression par exemple.

Nous nous baserons sur une fréquence d'actualisation moyenne calculée à partir de la vitesse de déplacement, soit une mise à jour de la position tous les 50 mètres. La vitesse

de déplacement utilisée est de 5 km/h soit une actualisation toutes les 36 secondes.

Actuellement l'actualisation de la position entraîne un trafic d'à peu près 250 octets. La mise à jour de la liste d'amis dépend quand à elle du nombre de contacts, mais d'une manière générale pèse 200 octets plus 55 octets par contact. Dans le cas d'une utilisation moyenne, en mouvement, avec 5 contacts dont 2 actifs le trafic horaire s'élève donc à 56 kilo octets. D'après les tarifs actuels des opérateurs qui varient de 0.01€ à 0.15€ par dizaine de kilo octets, le coût horaire peut aller de 0.055€ à 0.825€. Ce coût, même s'il est très faible, représente une barrière suffisante pour dissuader le grand public d'utiliser de telles applications.

Pour résumer, un utilisateur d'une formule prépayée chez un opérateur français devra supporter un coût horaire de 0,825€ alors qu'un utilisateur ayant un abonnement comprenant 1GO de données mensuelles supportera un coût horaire de 0,055€ (calculé sur la base du volume maximal et du prix de l'abonnement). Ces chiffres semblent misérables mais l'application devant fonctionner en permanence lorsque l'utilisateur souhaite être connecté à son réseau cela peut très vite devenir exorbitant. Pour deux heures d'utilisation quotidienne cela peut revenir à 49,50€ pour un mois d'utilisation.

L'optimisation du trafic est donc une phase très importante qui doit passer d'abord par le calcul des fréquences de rafraîchissement idéales puis par des techniques de compression des données échangées. C'est un des facteurs qui nous a fait utiliser un protocole réduit plutôt que le protocole **XMPP** (voir 3.5.3).

Dans le futur l'application pourrait migrer vers le protocole **XMPP** de « Jabber » qui implémente la plupart des fonctionnalités de présence et d'échange d'informations dont MoSoMuSo a besoin couplé à une compression et à une optimisation des données transférées.

#### 4.8.5 Coût en terme de consommation d'énergie

L'aspect financier n'est pas le seul à tendre en faveur de brefs échanges réguliers, la connexion 3G à partir d'un téléphone mobile est très coûteuse en énergie (mais ne sont qu'une partie de la consommation, le WiFi et le GPS sont également gourmands). C'est pourquoi le protocole de communication réseau fonctionne sans être en permanence connecté. Les chiffres exacts de consommation des téléphones ne sont pas communiqués mais, par exemple, les chiffres constructeur du Nokia N95 avec une batterie 3.7 volt (V) 1100 miliampère-heure (mAh) annoncent une perte d'autonomie de 25% en communication et de 11% en veille en mode **3G** par rapport au mode **GSM** pur.

En pratique, le fonctionnement en mode **3G** (ou bi-bande pour bénéficier du **GSM** dans les zones non couvertes) réduit le temps de veille à une journée et en communication le téléphone tombe en panne de batterie après quelques heures seulement.



# Chapitre 5

## Réalisation

Nous présenterons maintenant ce qui a été réalisé lors de ce stage. Nous commenceront par les détails techniques de la plateforme choisie et un bref comparatif avec ses concurrents puis passerons à l'application et certains de ses détails d'implémentation.

### 5.1 Choix techniques

La plateforme choisie l'a été à partir des besoins des différents partenaires. Le choix du téléphone plutôt que du Pocket PC réside dans plusieurs choses. Tout d'abord la relation étroite qu'entretiennent Clicmobile avec Nokia permet un support technique rapide et de grande qualité. D'autre part les téléphones Nokia sous Symbian OS semblaient plus simple à aborder au niveau développement. Symbian OS est un système d'exploitation dédié plus simple d'approche et beaucoup mieux documenté. Sony CSL avait également commencé à travailler sur plateforme Nokia.

Le fait que le téléphone mobile soit plus adapté à ce genre d'application a également joué, les Pocket PC sont vus par le public comme des outils professionnels et les téléphones sont maintenant devenus très communs. Par contre le Pocket PC que nous avons pu tester possédait les mêmes caractéristiques que le Nokia N95 et aurait très bien pu servir à la plateforme de test. Les seuls défauts que nous avons pu lui trouver est qu'il consomme très vite sa batterie et que le système agit comme une boîte noire, le réseau et le GPS semblaient très peu accessibles.

Pour information, un milliard de téléphones sous Symbian OS seraient en circulation aujourd'hui dans le monde (Source Symbian Ltd. mars 2007) et Nokia aurait vendu près de 100 millions de téléphones de type "smartphone" en 2006, en hausse de 50% par rapport à 2005.

Ainsi nous avons choisi le Nokia N95<sup>1</sup> pour son équipement très complet et pour ses performances annoncées. Ce téléphone fait partie des meilleurs de sa gamme de par son équipement (voir le comparatif 5.1). La présence d'un partenariat au sein du projet avec Nokia a également influencé le choix, cela assurait un support technique de qualité dans un délai raisonnable. Cela nous a permis de disposer d'un téléphone encore au stade de prototype à l'époque et de tester ses performances.

#### 5.1.1 Caractéristiques de la plateforme choisie

Le Nokia N95 est basé sur une architecture Texas Instruments *OMAP 2420* (décrite en annexe). Cette architecture est basée sur un microprocesseur **ARM11** à 330 Mhz, couplé

---

<sup>1</sup><http://www.forum.nokia.com/devices/N95>

|              | iPhone                  | N95                        | HTC P3300                 |
|--------------|-------------------------|----------------------------|---------------------------|
| Écran        | 3,5"                    | 2,6"                       | 2,8"                      |
| Résolution   | 320x480                 | 240x320                    | 240x320                   |
| Input method | écran tactile           | clavier                    | écran tactile             |
| Système      | Apple Mac OS X          | Symbian OS 9.2/S60 3.1     | Windows Mobile 5          |
| Stockage     | 4 ou 8 GO               | microSD                    | microSD                   |
| GSM          | quadribande             | quadribande et W-CDMA 2100 | quadribande               |
| Donnée       | EDGE                    | GPRS, EDGE, HSDPA          | EDGE                      |
| Connectivité | 802.11b/g, Bluetooth    | 802.11b/g, USB, Bluetooth  | 802.11b/g, USB, Bluetooth |
| Photo        | 2 méga pixels           | 5 méga pixels              | 2 méga pixels             |
| Autre        | iPod, compatible iTunes | Vidéo, GPS, radio FM       | GPS, radio FM             |

TAB. 5.1 – Comparatif Apple iPhone, Nokia N95, HTC P3300 Premium

à un processeur graphique PowerVR MBX de la société Imagination Technologies<sup>2</sup>.

### 5.1.2 Choix du langage

Le développement devait être rapide dans un premier temps pour permettre d'évaluer un premier prototype. C'est pourquoi, Python a été choisi plutôt que le C++ pour sa facilité et pour sa simplicité sur le système Symbian. En effet, le système Symbian OS introduit différents paradigmes et spécificités propres au monde de l'embarqué et le développement prend plus de temps alors que Python n'a subi que quelques ajouts de modules lui permettant de supporter le matériel spécifique au téléphone mais les paradigmes de programmation restent les mêmes.

Mais, bien que le C++ induise un temps de développement plus long, j'ai développé plusieurs outils dans ce langage du fait de l'impossibilité de le faire en Python.

## 5.2 L'application

Le client **MoSoMuSo** se compose de différents modules comme en témoigne la modélisation 4.4. Les figures 5.1 donnent un aperçu de l'interface graphique de l'application.

L'ensemble de l'application a été réalisée en utilisant le modèle de développement Modèle-Vue-Contrôleur mais nous détaillerons ici plutôt les modules fonctionnels composant l'application.

### 5.2.1 GPS

Le module d'acquisition des données GPS existant déjà lors de mon arrivée mais n'utilisait pas le GPS interne du téléphone car Python ne possédait pas de module d'interface avec le serveur de positionnement disponible dans le système qui est la seule porte d'accès à ce matériel. J'ai donc réalisé un module Python en C++ offrant une interface au serveur de positionnement ce qui permet d'utiliser le GPS interne et n'importe quel autre matériel de positionnement externe en Bluetooth ou un système de positionnement par le réseau si l'opérateur le propose (service généralement payant). Le module a été proposé à la communauté Python pour S60 [29] et sera peut être inclus dans une future version de la distribution **PyS60**.

<sup>2</sup><http://www.imgtec.com>



FIG. 5.1 – Interface graphique de l'application

Le module PyS60 d'interfaçage avec le serveur de positionnement de Symbian OS est disponible publiquement sur le dépôt de sources géré par Mercuriel [30] à l'adresse suivante : <https://air.mosomuso.com/pubhg/pyext/>. Le module est documenté et compile avec la version 1.3.21 de **PyS60**. Des scripts de création de l'archive d'installation sur le téléphone sont également disponibles.

### 5.2.2 Boussole

Le module de boussole est une vue qui permet de centraliser l'ensemble des données du réseau social et de permettre à l'utilisateur de les visualiser puis de les utiliser pour naviguer. La boussole a été créée comme une boussole standard avec une flèche indiquant la position de la cible par rapport au mouvement actuel de l'utilisateur. Elle se base sur le fait que l'utilisateur tient son téléphone dans le sens "logique" car il n'y a pas de capteur dans le N95 permettant de connaître l'orientation du téléphone. La flèche se base sur un calcul trigonométrique à partir de l'angle effectué par le déplacement par rapport à la ligne droite coupant le téléphone de haut en bas.

### 5.2.3 Synchronisation serveur

Le module de synchronisation serveur fonctionne en arrière plan et se charge de la mise à jour de la localisation (obtenue du module GPS) et de l'actualisation de la liste d'amis. La liste d'amis est mise à jour sur le serveur (si un utilisateur est ajouté par exemple) afin de garder un réseau social complet. La liste d'états est ensuite récupérée pour que changements d'état des autres utilisateurs soient répercutés dans l'interface graphique.

### 5.2.4 Serveur MoSoMuSo

Le serveur **MoSoMuSo** se compose d'un serveur dédié faisant fonctionner l'application écrite en Python avec le framework Django. Il se compose d'un site web permettant l'inscription des utilisateurs et les opérations de changement de mot de passe et de visualisation de la liste d'amis. Mais le site est surtout dédié à l'inscription, ensuite tout est fait à partir de l'application mobile qui se charge des mises à jour de la liste d'amis. La communication entre l'application et le site se fait par HTTP. Cette méthode à l'avantage de passer facilement les limitations introduites par les proxies de l'opérateur.

### 5.2.5 Composants simulés

Le système GPS ne fonctionne qu'en extérieur et est plutôt lent lors des tests c'est pourquoi j'ai développé un module de remplacement du GPS afin de pouvoir tester l'application sans être tributaire du temps de synchronisation ou du signal. De plus pour tester le mouvement pendant la phase de développement cela m'aurait obligé à bouger vraiment ce qui n'est pas envisageable. De ce fait, j'ai développé une interface graphique qui se compose d'une page HTML dotée d'une composante dynamique basée sur le principe AJAX. Celle ci permet d'émuler un récepteur GPS pour 4 client et de les faire bouger dans l'espace. La zone de déplacement représentée par le rectangle vert clair sur la figure 5.2 représente Paris et chaque mobile peut être déplacé dans cette zone. Le système souffre d'une précision très faible mais a l'avantage de permettre une démonstration en intérieur de façon claire. C'est un outil essentiel pour la conception et le prototypage.

Le système se base sur un module Python incorporé dans l'application et activé grâce au fichier de configuration qui effectue des requêtes HTTP vers le serveur **MoSoMuSo**. Le second élément du système est le composant répondant aux requêtes HTTP sur le serveur, c'est une interface à la base de donnée qui contient les positions. Le troisième élément est la page HTML qui permet de changer les positions et qui se charge d'effectuer des requêtes de mise à jour de la base de donnée sur le serveur.

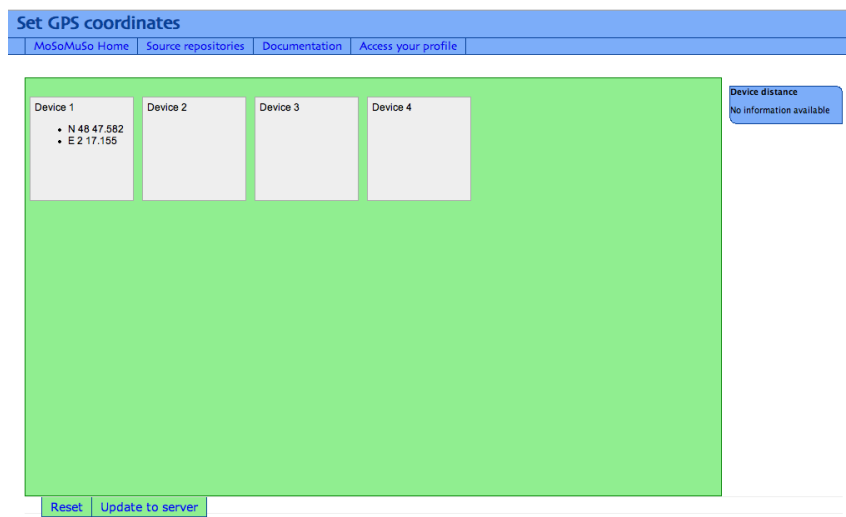


FIG. 5.2 – Simulateur GPS



## 5.3 Tests

La validation de la plateforme passe également par des tests qualitatifs, ceci pour s'assurer qu'elle ne limitera pas les performances de notre application et ne la rendra pas inutilisable pour l'utilisateur moyen. De plus, de bonnes performances générales nous donnerais la possibilité d'effectuer les opérations de traitement audio directement sur la plateforme et ainsi économiser en transfert réseau. Nous verrons dans cette partie les différents tests que nous avons effectué et essayerons de les mettre en parallèle pour évaluer plus objectivement la plateforme choisie.

### 5.3.1 Latence et portée wifi

La portée WiFi sur la plateforme choisie impacte la distance minimale à laquelle deux utilisateurs doivent se tenir pour interagir, c'est donc un facteur important et déterminant dans la navigation. De ce fait, lors de nos tests nous avons effectué un test de portée WiFi et avons découvert que les Nokia N95 étaient capable de communiquer jusqu'à une distance de 90 mètres. Des tests de latence à différentes distances seront fait dans le futur pour évaluer la faisabilité et les différents paramètres à prendre en compte lors de l'implémentation du streaming musical entre deux mobiles. Cette valeur impactant sur la durée gardée en mémoire (technique de *buffering*) afin de ne pas impacter sur la qualité d'écoute.

### 5.3.2 Traces de déplacement

Lors du *Mobile Music Workshop* nous avons équipé l'application d'un ensemble de fonctions enregistrant les actions des utilisateurs. Nous avons donc créé une base de traces d'utilisation sur une période d'une demi-heure. Ceci va nous permettre maintenant d'améliorer, d'une part, la qualité de l'interface graphique et d'autre part la navigation à l'aide de la boussole. En effet, les retours des utilisateurs nous indiquent que la boussole est destabilisant car elle oscille trop et a tendance à être complètement perdue lors de phases ou les utilisateurs ne bougent pas (pour discuter par exemple ou admirer le paysage). Ces traces se composent des positions relevées, de la position de la "cible", des fréquences de rafraîchissement qui sont adaptés à la vitesse, des clics dans les menus et de l'angle affiché par la boussole. Cela va nous permettre de rejouer des scénarios et de perfectionner l'algorithme de navigation.

### 5.3.3 Calcul flottant

La vitesse de calcul flottant est également déterminante en traitement du signal. **MoSoMuSo** requiert de traiter des fichiers musicaux, par conséquent la plateforme choisie doit être performante de ce point de vue. Mais la vitesse ne fait pas tout, en effet un calcul rapide mais comportant une trop grande approximation ne serait pas utile.

Les tests suivants ont été réalisés sur le Nokia N95 dont les caractéristiques sont décrites en 5.1.1.

Les tests ont été effectués à l'aide du programme *ffbench* [31] qui effectue une transformée de Fourier rapide d'une matrice carrée de 256 x 256 nombres complexes (flottants en double précision) suivie par la transformée inverse. 20 itérations sont faites dans ce comparatif. Après le test les résultats sont comparés aux valeurs connues du résultat afin de vérifier la précision de calcul. Le Nokia N95 n'a pas fait d'erreur sur le calcul. Il se place dans la moyenne par rapport à sa puissance.

| Temps (s)   | Système testé   |
|-------------|---|
| 17.25       | Silicon Graphics <i>Indigo</i> <sup>2</sup> , MIPS R4400, 175 Mhz, IRIX 6.5 |
| 14.10       | Dell Dimension XPS R100, Pentium II 400 MHz, Windows 98                     |
| 10.70       | Hewlett-Packard Kayak XU 450 Mhz Pentium II, Windows NT 4.0sp3              |
| <b>7.31</b> | <b>Nokia N95, ARM11 330 MHz, Symbian OS 9.2</b>                             |
| 5.09        | Sun Ultra 2, UltraSPARC V9, 300 MHz   |
| 0.846       | Dell Inspiron 9100, Pentium 4, 3.4 GHz                                      |

TAB. 5.2 – Comparatif de performances avec *ffbench*

### 5.3.4 Performances globales

Les chiffres donnés en 5.2 sont à mettre en relation avec ceux données par des application de “benchmarking” utilisées par les professionnels comme *SPMARK*<sup>04</sup> de la société Futuremark qui est spécialisée en test de plateformes mobiles. Les tests effectués sur ce type de plateformes (les “Smartphones” en particuliers) sont adaptés au contexte et sont très différents des tests qu’il est possible de voir dans les magazines d’informatique à propos des derniers processeurs ou cartes graphiques.

Les tests faits par *SPMARK*<sup>04</sup> [32] tiennent compte de la manipulation du calendrier, de la manipulation d’images (redimensionnement d’images et compression JPEG en particulier), de la manipulation vidéo sur les appareils comportants une caméra (compression, décompression), accès à la mémoire et aux disques, affichage 3D pour les appareils comportant un accélérateur 3D matériel.

Les résultats obtenus en 5.3 confirment les chiffres donnés en 5.2. Ils montrent bien le Nokia N95 est performant en calcul 3D grâce à son matériel dédié. Le Nokia N95 est placé parmi les meilleurs téléphones, même si le Nokia N93 le surpasse dans le cas de la 3D, le N95 est un téléphone bien plus complet et donc intéressant pour [MoSoMuSo](#).

|         | Nokia N93            | Nokia N95            | SonyEricsson M600i |
|---------|----------------------|----------------------|--------------------|
| OS      | Symbian OS v9.1      | Symbian OS v9.2      | Symbian OS v9.1    |
| CPU     | ARM 206MHz           | ARM 320 MHz          | ARM 208 MHz        |
| 3D      | PowerVR MBX avec VGP | PowerVR MBX avec VGP | PowerVR MBX Lite   |
| Score   | 5626                 | 6811                 | 2155               |
| 3D      | 8914                 | 6825                 | 2496               |
| Système | 2389                 | 2196                 | 1216               |

TAB. 5.3 – Comparatif de performances avec *SPMARK*<sup>04</sup>

### 5.3.5 Vitesse de lecture en mémoire de masse

Un des goulot d’étranglement est la lecture de données en mémoire. Notre prototype utilisait la lecture en mémoire au moment du transfert de fichier et la lenteur de la lecture s’est ressentie sur le temps de transfert. C’est pourquoi nous avons fait des tests de performance de lecture pour évaluer la meilleure méthode pour lire un fichier (carte mémoire ou disque en RAM, taille de bloc).

Voici un récapitulatif des résultats sur la figure 5.4. La carte mémoire utilisée est une MicroSD Sandisk UltraII.

| Disque        | RAM    |       | Flash  |          |
|---------------|--------|-------|--------|----------|
|               | 1 Mo   | 10 Mo | 1 Mo   | 10 Mo    |
| Buffer 1 ko   | 0.3075 | 3.048 | 1.524  | 14.41275 |
| Buffer 100 ko | 0.124  | 1.237 | 1.167  | 11.67875 |
| Buffer 1 Mo   | 0.123  | 1.216 | 1.1615 | 11.616   |

TAB. 5.4 – Performance de la lecture mémoire (s)

## 5.4 Dissémination

La participation aux conférences et la publication d’articles est un travail important dans le domaine de la recherche car il permet de faire connaître son travail et de rencontrer d’autres personnes ayant les mêmes problématiques. L’écriture de l’article “Social Mobile Music Navigation Using The Compass” [6] basé sur mon implémentation et sur le concept développé dans le projet a permis de se rendre compte que le concept de la boussole comme outils pour la création du réseau était une idée intéressante d’autres personnes et surtout de rencontrer d’autres personnes faisant le même type d’applications lors du *Mobile Music Workshop* et donc d’échanger des idées et des pistes de recherche afin d’améliorer l’application.

L’article est donné en annexe.

## 5.5 Problèmes rencontrés

Le développement a été ponctué de nombreux problèmes dûs principalement à la plateforme. En effet, même si la plateforme était la plus intéressante, elle était encore trop jeune pour être complètement stable. D’autres problèmes ont été introduits par le langage de programmation, Python qui reste encore à l’état de projet en développement. Nous allons détailler cela maintenant.

### 5.5.1 Droits d’accès de la plateforme

La plateforme choisie, le téléphone mobile Nokia N95 est une plateforme relativement protégée grâce aux mécanismes de sécurité introduits par la version 9 de son système d’exploitation Symbian OS. Celle-ci nécessite que les applications soient signées par un certificat dans lequel sont intégrés les droits donnés aux applications.

Ce système est une très bonne chose pour éviter la prolifération d’applications gratuites disponibles sur Internet qui introduiraient des “chevaux de troie” dans les téléphones mobiles, ou pire encore, qui utiliseraient le carnet d’adresse pour envoyer des messages non sollicités.

### Point de vue développement

L’inconvénient de ce système est que le développeur n’a plus de liberté et chaque recompilation nécessite la signature de l’application puis, lors de l’installation, la vérification de cette signature par le téléphone. Une solution est de développer sur simulateur mais cela reste handicapant pour les tests réels.

Les certificats peuvent être générés mais un certificat non signé par une autorité reconnue par le système n’aura pas la possibilité d’accorder tous les droits possibles. Ainsi le plus gros frein au développement est l’obtention d’un certificat. En effet, ces certificats sont payants et nécessitent de suivre une formation spécifique. Symbian Signed [33], l’entité

gérant et centralisant la certification offre des certificats développeurs gratuits mais ceux-ci sont bloqués à un téléphone en particulier grâce à son **IMEI**. Ceci rend très longue la phase de tests et d'installation sur plusieurs téléphones. De plus, ces certificats ne donnent pas tous les droits, les droits limités aux opérateurs ne sont jamais accessibles.

Certains droits doivent être explicitement demandés et nécessitent de faire une demande justifiée auprès de Symbian Signed. C'est le cas pour certains droits ayant attrait aux DRM, mais aussi à l'écriture disque et, ce qui est plus contraignant pour notre application, au réseau. Ainsi il n'est pas possible d'émettre des messages ICMP sans ce droit.

Il est également impossible d'utiliser le système interne de positionnement sans le droit "location". C'est un serveur applicatif qui centralise les différents moyens de positionnement (GPS interne au téléphone, GPS externe Bluetooth, réseau de l'opérateur) et qui offre une interface unifiée d'accès à l'information.

### Point de vue distribution de l'application

Il est impossible de distribuer une application signée avec un certificat de développement car elle est bloquée à un téléphone. Par conséquent, si l'application utilise des droits non basiques il est obligatoire de passer par Symbian Signed. Symbian Signed fera des tests de sécurité sur l'application avant de la certifier, ainsi le moindre changement (nouvelle version, recompilation, ajout de fonctionnalités) requiert de repasser par la phase de tests (et ainsi de payer ce service). Il est possible de faire certifier et signer un "freeware" gratuitement mais cette étape prend plus de temps que pour le service payant.

### Gestion de la sécurité dans Symbian OS

Seulement 60% des API sont disponibles sans nécessiter de signature avec un droit en particulier, les autres requièrent de signer l'application et donc, obligent le programmeur à disposer du niveau suffisant pour disposer du droit.

Il existe trois types de droits, les premiers peuvent être utilisés avec un certificat auto-généré et concernent l'utilisateur et son environnement comme par exemple "location" qui donne l'accès au serveur de positionnement ou "UserEnvironnement" qui permet de lire les données de l'utilisateur. Un certificat auto-généré entraînera par contre un avertissement de l'utilisateur au moment de l'installation.

Ensuite viennent les droits plus critiques pour l'environnement de l'utilisateur qui nécessitent de faire une demande écrite auprès de Symbian Signed qui étudiera la demande. Cela concerne l'accès aux informations de sécurité comme le contenu de la carte **SIM** ou l'accès aux périphériques.

Certaines fonctionnalités sont limitées aux opérateurs comme la gestion des DRM, le control bas niveau du réseau (envoi de paquets ICMP, utilisation de la pile GSM). Ces droits peuvent être accordés de manière exceptionnelle sur demande et justification auprès de Symbian Signed et de Nokia.

Un détail des droits disponibles se trouve en annexe.

#### 5.5.2 Limitations de Python

Python pour série 60 **PyS60** a été lancé par Nokia en 2004-2005 mais n'a que très peu évolué jusqu'en 2006 parce que, d'une part, le projet n'était pas *open sources* mais aussi parce que les smart phones n'avaient pas encore percé le marché comme aujourd'hui. Depuis fin 2006 les choses se sont accélérées et le projet est très actif, néanmoins PyS60 est encore basé sur une ancienne version de Python (2.2). Depuis le 1er juin 2007, *Pymalloc*, un

allocateur mémoire plus performant a été importé de Python 2.5 vers **PyS60** et la version de **PyS60** 1.3.22 gagne en rapidité. Des gains de performance de 300% ont été constatés par les développeurs de **PyS60** sur certaines fonctions.

De plus cette nouvelle version apporte des optimisations spécifiques à la plateforme OMAP 2420 du N95 ce qui permet aussi de gagner en performances.

D'autres choses comme la primitive de texte qui permet d'afficher du texte ne fonctionnait pas sur les téléphones de la série 3 (Symbian OS 9 et supérieur) ce qui rendait l'utilisation de Python quasi impossible pour des applications graphiques ou comportant une interface graphique. Ceci avait été corrigé par un patch temporaire que MoSoMuSo utilisait mais qui rendait les applications incompatibles avec les anciennes version de PyS60.

### Implémentation des sockets

Les problèmes les plus importants et les plus difficile à localiser ont été ceux touchant l'implémentation des **socket**. En effet certaines fonctions ne sont que partiellement implémentées et ne font l'objet d'aucun avertissement ou erreur. Leur comportement est aléatoire, comme la fonction *select()* qui n'est implémentée qu'à un tiers et qui ne fonctionne pas sur le résultat de la fonction *accept()* ceci rend la création d'une boucle d'attente de connexion entrante bloquante. Ce bug a été signalé le 18 avril 2007 par un membre de l'équipe de développement mais n'a pas encore été corrigé.

De plus, une socket ouverte ne peut pas être partagée entre plusieurs threads, combiné au problème de la fonction *accept()* cela rend impossible la réalisation d'un serveur pouvant accepter de multiples connexions entrantes. Cette limitation devrait être corrigée dans une future version de **PyS60**

### Sockets et Bluetooth

On a constaté que l'utilisation de plusieurs sockets TCP ou INET en parallèle avec des sockets Bluetooth conduit parfois à la perte de la communication avec le périphérique Bluetooth. Ce comportement est assez aléatoire et semble dépendre de l'utilisation du téléphone, il pourrait donc ne pas être directement lié à l'implémentation des sockets. À ce jour nous n'arrivons pas à le reproduire de façon déterministe.

### Stabilité de l'interface graphique

Les éléments graphiques typiques tels que les listes comportant des icônes ("*Listbox*") sont très instables lorsqu'elles sont utilisées avec des éléments sur lesquels il est possible de dessiner (les "*Canvas*"). Ceci m'a obligé à repenser entièrement l'interface et à n'utiliser que des images pour faire l'affichage. Ceci implique que le fonctionnement événementiel de l'interface a été perdu au bénéfice d'une boucle principale effectuant les calculs et le rafraîchissement de l'affichage.

### Droits liés à l'interpréteur Python

Chaque service qui est utilisé nécessite un certain nombre de droits donnés grâce à la signature de l'exécutable comme on l'a décrit en 5.5.1. De ce fait, Python nécessite d'être signé avec les droits suffisants pour chaque service qu'il utilise. Nous avons été confrontés à ce problème lors de la réalisation du module qui utilise le service de positionnement. En effet l'interpréteur PyS60 est distribué avec le strict minimum de droits car il est signé avec un certificat auto-généré. Ceci nous a obligé à signer l'interpréteur avec un certificat développeur obtenu chez Symbian Signed.

Ce problème est à double tranchant, en effet un interpréteur Python disposant de droits trop importants permettrait à n'importe qui d'outrepasser les mécanismes de Symbian OS et d'accéder à des informations confidentielles voire de passer des communications onéreuses pour le propriétaire du téléphone. De ce fait, l'interpréteur distribué ne peut pas être doté de droits supplémentaires. Cela nous obligera de distribuer une application packagée embarquant son interpréteur Python et signé avec des droits suffisants.

### 5.5.3 Hétérogénéité des partenaires

Bien que le projet soit né d'une volonté commune des trois partenaires, ceux-ci n'en attendent pas les mêmes choses. En effet, au LIP6 le but est de travailler sur de nouveaux protocoles et de publier ces recherches. Clicmobile a des attentes complètement différentes car il attend la conception d'une application innovante afin de pouvoir en dériver des produits ou services commerciaux. Entre ces deux aspects, Atau Tanaka chez Sony CSL privilégie la recherche mais souhaite utiliser la plateforme pour y intégrer ses travaux.

Il n'a donc pas été facile de concilier les différentes attentes et de contenter tous les membres du projet.

## Chapitre 6

# Bilan

Mon travail au sein du projet a permis de concrétiser les idées qui avaient germé au sein de l'équipe et d'en explorer d'autres. L'objectif de développer un protocole de communication pour téléphone mobile ou Pocket PC n'est pas encore finalisé, mais le prototype réalisé en est la première brique permettant de valider nos choix de conception. Grâce à ce travail nous avons pu réaliser une première expérimentation du concept du projet et ainsi nous rendre compte de potentiel qu'il avait et de l'intérêt des gens pour lui. En effet les premiers retours sont très positifs et notre concept de boussole fonctionne. En effet, les tests effectués auprès de volontaires ont démontré que trois quarts des gens sont prêts à bouger pour suivre la direction indiquée en faisant confiance à l'application lorsqu'il y a une incitation. Mais par contre l'oscillation trop importante a destabilisé la quasi totalité des gens qui ont été perdus à un moment donné. Par contre la réactivité générale du téléphone, de l'application et de la navigation dans les menus ont semblé être largement suffisants et plaire aux testeurs.

De notre coté, la plateforme Nokia N95 s'est montré un peu moins ouverte et simple d'approche que prévu initialement mais elle confirme sa qualité et sa puissance. En effet, l'autonomie en utilisation du WiFi s'est montrée supérieure à deux heures et la portée a atteint les 90 mètres en environnement urbain dégagé. Pour le projet elle représente maintenant la meilleure solution pour passer à une expérimentation à grande échelle car elle pourra très facilement être utilisée par des volontaires. De plus, au vu de sa puissance, il semble qu'elle soit tout à fait capable de supporter le traitement de la musique en temps réel et ainsi que la musique malléable pourra y être implémentée.

Le choix du langage est un peu plus discutable à cause des différentes limitations que nous avons rencontré lors du développement. Mais une nouvelle version de **PyS60** (1.3.22) le 1er juin 2007 apporte de grandes améliorations de stabilité et de performance. Et les problèmes que nous avons rencontrés sont en bonne voie pour être corrigés.

Par ailleurs, le travail d'un chercheur se compose également d'une part de communication et de dissémination de l'information et les différents événements auxquels nous avons participé ont contribué à la visibilité du projet. Ceci est une bonne chose car nous sommes aujourd'hui en contact avec des gens qui travaillent sur les mêmes problématiques et pouvons échanger des informations. Notre participation à deux workshops "interactions sociales par la musique" chez Sony CSL avec le laboratoire InSitu de l'INRIA nous a également permis de nous rendre compte des modèles d'utilisation des périphériques musicaux et des besoins ressentis par les utilisateurs.

J'ai donc apporté mes compétences en ingénierie dans l'équipe et grâce à la collaboration avec Clicmobile et Sony CSL nous avons pu créer une application intéressante scientifiquement mais également commercialement. Cette collaboration a été très enrichissante.

Ayant été embauché avec un contrat d'un an, il me reste maintenant sept mois pour concrétiser l'objectif premier, c'est à dire créer une application capable de créer des interactions spontanés entre ses utilisateurs, comme le font les être humains lorsqu'ils se rencontrent. Nous allons présenter quelques pistes qui seront explorées dans mes futurs travaux.

Les premiers travaux de conception nous dirigent vers un nouveau concept : "Incentive Mobile Networks" qui se base sur le routage incitatif. En effet, un utilisateur pourrait être invité à bouger s'il veut garder la connectivité ou une qualité de service acceptable. Ceci dans le but de conserver la connectivité lui-même ou alors de connecter d'autres partie du réseau.

D'autre part, le projet est maintenant doté d'un bon outil de mesure des utilisations d'un téléphone mobile. C'est pour cela que l'application va être consolidée et dotée d'autres mécanismes d'enregistrement des usages, tels que le temps de communication, le nombre de messages texte envoyés et reçus ou les antennes GSM sur lesquelles le téléphone s'enregistre. Ceci va nous permettre de distribuer une vingtaine de téléphones mobiles à des volontaires chargés de les utiliser et de les avoir sur eux. Nous pourrions ainsi collecter un nombre important de traces de mobilité afin d'étudier les modèles de mobilité des gens. Ce genre de données ne sont pas disponibles car tous les travaux se basent sur des populations homogènes et en général travaillant dans les mêmes lieux et étant sensiblement de la même classe sociale. Le but pour nous est de collecter des données venant de sources hétérogènes pour pouvoir les analyser. Ce serait également une bonne contribution pour la communauté scientifique travaillant sur la mobilité.

Parallèlement, la boussole pourrait être améliorée grâce au filtrage des positions afin de la stabiliser. Même si ce n'est pas ma mission principale, c'est un aspect intéressant du projet que nous souhaiterions développer et finir afin d'en présenter une version stable. D'autre part la boussole fait partie des outils de contrôle du réseau auto-organisable, elle garde donc tout son sens au sein du projet. En effet, dans l'approche choisie d'inciter les utilisateurs à bouger pour provoquer la création d'un réseau il est important de proposer un outil efficace pour le faire.

**Bilan personnel :** ce stage m'a permis de mettre en pratique mes connaissances que j'ai acquies au fil de mes études à EPITA. La spécialisation Calcul Scientifique et Image ne m'a pas apporté de connaissances particulières qui touchaient le domaine dans lequel j'ai travaillé mais plutôt un ensemble de méthodes que j'ai pu appliquer. Même si j'ai mis beaucoup de temps à m'adapter et à démarrer au début je suis aujourd'hui plus à l'aise au sein du LIP6 pour pouvoir mener à bien les tâches qui m'attendent.

Après 5 mois, je me rend compte que le travail n'est pas aussi rapide en entreprise qu'à l'école car il est souvent lié à d'autres facteurs comme les réunions ou l'attente de pièces faites par les partenaires. Aujourd'hui je pense que mon avancement n'a pas été aussi rapide qu'il aurait pu l'être à cause d'hésitations, aussi bien dans les choix que dans la prise d'initiatives. Cette expérience a été enrichissante dans le sens où j'ai été chargé du développement en coordonnant les différents besoins de chercheurs qui travaillent sur des



---

domaines complètement différents et qui souhaitent retrouver différentes fonctionnalités dans l'application.

Cette expérience m'a également permis de travailler au sein d'un des plus grand laboratoire de recherche français avec tous les avantages que cela implique, la disponibilité de gens très compétents dans de nombreux domaines et le nombre important de travaux très différents. Ainsi mon stage se termine sur un bilan plutôt positif, les résultats obtenus lors de mon travail sont concluants et la suite promet d'être encore plus intéressante et se terminera par des expérimentations grande échelle du protocole réalisé.



# Glossaire

|              |  |
|--------------|--|
| 3G           | Terme désignant la troisième génération de protocole téléphonique mobile permettant une connexion donnée à haut débit, <a href="#">38</a> , <a href="#">43</a>   |
| ad hoc       | réseau sans fil créé dans un but précis capable de s'organiser sans infrastructure définie préalablement, <a href="#">24</a> , <a href="#">33</a>  |
| AOA          | Angle of arrival, <a href="#">31</a>   |
| ARM          | Processeurs basés sur une architecture RISC 32 bits tirant leur nom de Acorn RISC Machines. C'est une architecture simple et performante qui a été développée par la société ARM Limited et qui est largement utilisée dans nombre d'applications mobiles ou embarquées, <a href="#">32</a> , <a href="#">45</a> |
| Blue Jacking | Néologisme anglais de Bluetooth et "to jack" désignant l'envoi de messages non sollicité sur des appareils utilisant la technologie Bluetooth, <a href="#">24</a>  |
| BSC          | Base Station Control, <a href="#">27</a> , <a href="#">29</a>  |
| BSS          | Base Station Subsystem composée du BTS et du BSC, <a href="#">27</a>   |
| BTS          | Base Transceiver Station, <a href="#">27</a>   |
| DTN          | Delay Tolerant Network, <a href="#">35</a>   |
| E911         | Enhanced 911 est un service du numéro d'urgence sur le réseau téléphonique nord américain qui permet d'associer une adresse physique (ou une localisation) à un numéro lors d'un appel d'urgence., <a href="#">30</a>  |
| EDGE         | « Enhanced Data Rates for GSM Evolution » est une évolution du GSM assurant un débit de 384 kilo bit par seconde, <a href="#">29</a> , <a href="#">30</a>  |
| ETSI         | European Telecommunications Standards Institute, <a href="#">27</a>  |
| GPRS         | General Packet Radio Service, <a href="#">28–30</a>  |

|            |   |
|------------|---|
| GPS        | « Global Positioning System » est un système de positionnement mondial mis en place par le département de la défense américaine qui se base sur une flotte de 24 satellites orbitant à 20200km d'altitude pour permettre à tout objets équipé d'un recepteur adéquat d'obtenir sa position à 20m près (depuis 2000 et la fin de la dégradation volontaire de la précision du GPS), <a href="#">31</a> , <a href="#">37</a> , <a href="#">38</a> |
| GSM        | "Groupe Spécial Mobile" devenu "Global System for Mobile communication" en 1991, <a href="#">27</a> , <a href="#">28</a> , <a href="#">30</a> , <a href="#">43</a> , <a href="#">56</a>   |
| GTP        | GPRS Tunnelling Protocol, <a href="#">29</a>  |
| HSDPA      | "High-Speed Downlink Packet Access" est une amélioration de l'UMTS portant le débit descendant jusqu'à 3.6 Mb/s. C'est la 3.5G ou 3G+, <a href="#">30</a>   |
| IMEI       | International Mobile Equipment Identity, <a href="#">28</a> , <a href="#">51</a>  |
| IRCAM      | Institut de Recherche et Coordination Acoustique/Musique, <a href="#">25</a>  |
| LIP6       | Laboratoire d'informatique de Paris VIN, <a href="#">15–18</a> , <a href="#">56</a>   |
| MIDI       | Musical Instrument Digital Interface, <a href="#">25</a>  |
| modèle OSI | Le modèle OSI Le modèle d'interconnexion en réseau des systèmes ouverts de l'Organisation internationale de normalisation ISO est un modèle de communications entre ordinateurs. Il décrit les fonctionnalités nécessaires à la communication allant du matériel au logiciel en sept couches successives, <a href="#">29</a>  |
| MoSoMuSo   | Mobile Social Music Software, <a href="#">15–17</a> , <a href="#">19</a> , <a href="#">25</a> , <a href="#">35</a> , <a href="#">36</a> , <a href="#">38</a> , <a href="#">40</a> , <a href="#">41</a> , <a href="#">46–50</a>  |
| MoSoSo     | Mobile Social Software, <a href="#">23</a>  |
| MS         | Mobile Station désigne un terminal mobile connecté au réseau GSM, <a href="#">29</a>  |
| MSISDN     | Mobile Station ISDN Number, <a href="#">28</a> , <a href="#">37</a>   |
| NPA        | Network and Performance Analysis est un équipe du département "Réseaux" du LIP6, <a href="#">3</a> , <a href="#">16</a> , <a href="#">17</a>  |
| OLSR       | Optimized Link State Routing Protocol, <a href="#">33</a>   |

---

|        |   |
|--------|---|
| PSK    | Phase-shift keying soit “modulation par sauts de phase” désigne une famille de formes de modulations numériques qui ont toutes pour principe de véhiculer de l’information binaire via la phase d’un signal de référence (porteuse), <a href="#">29</a> |
| PyS60  | Python for S60 platform, <a href="#">46</a> , <a href="#">52</a> , <a href="#">53</a> , <a href="#">55</a>  |
| QoS    | Quality of Service, <a href="#">35</a>  |
| RIAM   | Réseau pour la Recherche et l’Innovation en Audiovisuel et Multimédia, <a href="#">3</a> , <a href="#">16</a> , <a href="#">18</a> , <a href="#">19</a>   |
| RTCP   | réseau téléphonique commuté public, <a href="#">27</a>  |
| RTSP   | Real-time streaming protocol, <a href="#">34</a>  |
| SGSN   | Serving GPRS Support Node, <a href="#">29</a>   |
| SIM    | Subscriber Identity Module, <a href="#">27</a> , <a href="#">52</a>   |
| socket | interface logicielle de connexion réseau qui peut être comparé à une prise de connexion qui permet de communiquer avec une application distante, <a href="#">53</a>   |
| SON    | Self Organizing Network, <a href="#">33</a>   |
| TDOA   | Time difference of arrival, <a href="#">31</a>  |
| TIMSI  | Temporary Mobile Station Identifier, <a href="#">28</a>   |
| UMTS   | ”Universal Mobile Telecommunications System” est une connexion à haute vitesse offrant des débits de 384 kb/s. C’est la 3G, <a href="#">29</a> , <a href="#">30</a>   |
| VoIP   | La voix sur IP est une technique qui consiste à communiquer vocalement sur un réseau IP, <a href="#">30</a>   |
| W-CDMA | Wideband Code Division Multiple Access, <a href="#">29</a>  |
| Wap    | Wireless Application Protocol définit une structuration des services internet adaptée aux mobiles (avec des contraintes de bande passante et de taille d’écran), <a href="#">24</a>   |
| XMPP   | « eXtensible Messaging and Presence Protocol » est le nom d’un protocole standard ouvert de l’IETF de messagerie instantanée basé sur les RFC 3920 à 3923, <a href="#">34</a> , <a href="#">43</a>  |

# Index

- ad hoc, 33
- AOA, 31
- Blue Jacking, 24
- BSC, 29
- Clicmobile, 17
- DTN, 35
- E911, 30
- EDGE, 29, 45
- ETSI, 27
- GPRS, 29, 45
- GSM, 27
- GTP, 29
- HSDPA, 30, 45
- IMEI, 28
- Jabber, 43
- LIP6, 15, 16
- MAX/MSP, 25
- MoSoMuSo, 16, 41
- MoSoSo, 23
- MSISDN, 28
- OLSR, 33
- propriété intellectuelle, 25
- PSK, 29
- Pure Data, 25
- PyS60, 52
- Python, 52
- Réseau auto-organisable, 33
- réseau social, 16
- RTCP, 27
- Self Organizing Network, 33
- SGSN, 29
- SIM, 27, 28
- SON, 33
- Sony, 17
- Sony CSL, 17
- Sony CSL, 13
- Symbian OS, 32
- TDOA, 31
- TIMSI, 28
- VoIP, 30
- W-CDMA, 29, 45
- Wap, 24
- XMMP, 43

# Bibliographie

- [1] “Présentation du lip6.” [Online]. Available : <http://www.lip6.fr>
- [2] L. Edwards and R. Barker, *Developping Series 60 Applications, A Guide for Symbian OS C++ Developers*. Pearson Education, 2004.
- [3] Nokia, “Nokia developer forum.” [Online]. Available : <http://www.forum.nokia.com/>
- [4] A. Agha, Pujolle, and Vivier, *Réseaux de mobiles & réseaux sans fil*. Eyrolles, 2006.
- [5] R. Meraihi and G. L. Grand, “Contrôle de topologie orienté stratégie basé sur des routeurs mobiles dédiés,” *Annales des Télécommunications*, vol. 61, no. 5-6, pp. 602–626, May, June 2006, iSSN 0003-4347. [Online]. Available : <http://perso.enst.fr/~legrand/Publis/Annales.pdf>
- [6] A. Tanaka, G. Valadon, and C. Berger, “Social mobile music navigation using the compass,” in *Proceedings of the 4th International Mobile Music Workshop*, May 2007, pp. 34–37.
- [7] “Myspace,” MySpace.com Inc. [Online]. Available : <http://www.myspace.com>
- [8] “Flickr,” 2007. [Online]. Available : <http://www.flickr.com/>
- [9] “Juice caster.” [Online]. Available : <http://www.juicecaster.com>
- [10] “Linkedin.” [Online]. Available : <http://www.linkedin.com>
- [11] “Viadeo.” [Online]. Available : <http://www.viadeo.fr>
- [12] “Meetic.” [Online]. Available : <http://www.meetic.fr/>
- [13] “Nokia sensor.” [Online]. Available : <http://www.nokia.com/sensor/>
- [14] “Mobiflirt.” [Online]. Available : <http://www.mobiflirt.com>
- [15] “Zune digital media player.” [Online]. Available : <http://www.zune.net>
- [16] M. Jacobsson, M. Rost, M. Håkansson, and L. E. Holmquist, “Push!Music : Intelligent Music Sharing on Mobile Device,” in *Adjunct Proceedings of UbiComp 2005*, Tokyo, Japan, 2005, demonstration. [Online]. Available : [http://www.viktoria.se/fal/publications/2005/Jacobsson\\_final\\_version\\_ubicomp05.pdf](http://www.viktoria.se/fal/publications/2005/Jacobsson_final_version_ubicomp05.pdf)
- [17] A. Bassoli, J. Moore, and S. Agamanolis, “tunA : Socialising Music Sharing on the Move,” in *Consuming Music Together : Social and Collaborative Aspects of Music Consumption Technologies*. Springer-Verlag, 2006, Book chapter. [Online]. Available : <http://web.media.mit.edu/~stefan/hc/publications/Bassoli04TunaUBICOMP.pdf>
- [18] A. Tanaka, “Malleable mobile music,” in *Adjunct Proceedings of the 6th International Conference on Ubiquitous Computing (UbiComp)*, 2004. [Online]. Available : <http://www.csl.sony.fr/downloads/papers/2004/tanaka-04c.pdf>
- [19] M. Puckette, “Pure data.” [Online]. Available : <http://puredata.info/>
- [20] F. C. Federation, “Enhanced 911 - wireless services,” 2007. [Online]. Available : <http://www.fcc.gov/911/enhanced/>

- [21] ilico.net, “Ootay,” 2007. [Online]. Available : <http://www.ootay.fr/>
- [22] “Gsm-tracker, free and user supported, mobile location service,” exoSyphen Studios, 2007. [Online]. Available : <http://www.gsm-tracker.com/>
- [23] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP : A Transport Protocol for Real-Time Applications,” RFC 3550 (Standard), July 2003. [Online]. Available : <http://www.ietf.org/rfc/rfc3550.txt>
- [24] A.-V. T. W. Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP : A Transport Protocol for Real-Time Applications,” RFC 1889 (Proposed Standard), Jan. 1996, obsoleted by RFC 3550. [Online]. Available : <http://www.ietf.org/rfc/rfc1889.txt>
- [25] H. Schulzrinne, A. Rao, and R. Lanphier, “Real Time Streaming Protocol (RTSP),” RFC 2326 (Proposed Standard), Apr. 1998. [Online]. Available : <http://www.ietf.org/rfc/rfc2326.txt>
- [26] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP) : Core,” RFC 3920 (Proposed Standard), Oct. 2004. [Online]. Available : <http://www.ietf.org/rfc/rfc3920.txt>
- [27] —, “Extensible Messaging and Presence Protocol (XMPP) : Instant Messaging and Presence,” RFC 3921 (Proposed Standard), Oct. 2004. [Online]. Available : <http://www.ietf.org/rfc/rfc3921.txt>
- [28] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [29] “Python for s60.” [Online]. Available : <http://pys60.sourceforge.net>
- [30] “Mercurial distributed source management tool for software developpers.” [Online]. Available : <http://selenic.com/mercurial>
- [31] J. Walker, “Fast fourier transform benchmark,” 1989. [Online]. Available : <http://www.fourmilab.ch/fbench/ffbench.html>
- [32] S. Niemelä, “Spmark04 for symbian os v9 whitepaper,” 2004. [Online]. Available : [http://www.futuremark.com/companyinfo/pressroom/companypdfs/SPMark04\\_Whitepaper.pdf?m=v](http://www.futuremark.com/companyinfo/pressroom/companypdfs/SPMark04_Whitepaper.pdf?m=v)
- [33] “Symbian signed.” [Online]. Available : <http://www.symbiansigned.com>