

UNIVERSITÉ DE MONTRÉAL

ADAPTABILITÉ D'APPLICATIONS MOBILES DANS
UN CONTEXTE D'APPRENTISSAGE

OUSMANE DIOUF
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE
MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AVRIL 2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-19295-5

Our file Notre référence

ISBN: 978-0-494-19295-5

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ADAPTABILITÉ D'APPLICATIONS MOBILES DANS
UN CONTEXTE D'APPRENTISSAGE

présenté par : DIOUF Ousmane

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. DESMARAIS Michel, Ph.D., président

M. PIERRE Samuel, Ph.D., membre et directeur de recherche

M. BOUDREAULT Yves, Ph.D., membre

REMERCIEMENTS

Je tiens tout d'abord à remercier de tout cœur mon directeur de recherche, monsieur Samuel Pierre, pour sa disponibilité, la pertinence de ses conseils et sa patience. Merci pour tout.

A tous les membres du LARIM (Laboratoire de Recherche en Réseautique et Informatique Mobile) de l'École Polytechnique de Montréal et à tous ceux qui ont contribué à l'aboutissement de ce travail de recherche.

J'exprime aussi toute ma reconnaissance à mes parents, frères, sœur et amis pour leur soutien moral indéfectible, ainsi que pour leurs encouragements constants.

Enfin plus que des remerciements, à mon épouse qui a su supporter mes sautes d'humeur, fin de semaines et vacances studieuses. Malgré la grande distance qui nous a séparés ces derniers mois, tu as su me remotiver quand le besoin s'en faisait sentir.

RÉSUMÉ

L'émergence des nouvelles technologies de l'information et des communications permet l'accès à un nombre sans précédent d'information en tout temps pour les étudiants. Cependant, le caractère de plus en plus nomade de ces étudiants, la diversité de leurs sources d'information, ainsi que les faibles capacités des appareils et liens de communication mobiles, font que les paradigmes actuels ne sont pas adaptés à l'apprentissage mobile.

En effet, les appareils mobiles présentent certaines contraintes liées à la taille de leur écran, leur clavier, leur vitesse de calcul, leur consommation d'énergie, ainsi qu'à la diversité de leur plate-forme d'exécution. Ceci rend l'accès à de l'information spécifique dans un contexte de mobilité difficile. La prochaine génération d'application mobile entend surmonter toutes ces limitations, en permettant aux utilisateurs d'avoir accès à de l'information concise et adaptée, même lorsqu'ils sont en mouvement.

Dans ce mémoire, nous proposons une architecture de service novatrice permettant aux étudiants mobiles de recevoir des données adaptées à leur contexte géographique, et aux caractéristiques des appareils mobiles qu'ils utilisent. A partir de cette architecture, nous proposons des modèles d'utilisation qui permettent de détecter l'utilisateur, le contexte d'utilisation, et aussi, l'accès à l'information adéquate par les étudiants. Notre modèle de service répond à la plupart des contraintes dictées par l'utilisation de plates-formes mobiles. La détection, l'identification et le transfert des données aux utilisateurs mobiles se font à l'aide du protocole Bluetooth s'exécutant sur une plate-forme J2ME, conçue pour s'exécuter sur plusieurs systèmes d'exploitation.

Nos résultats ont démontré que notre modèle d'application peut être implémenté sur plusieurs appareils mobiles dotés d'une plate-forme d'exécution Java. De plus, lorsque des stations de base Bluetooth sont placées à des endroits stratégiques à forte affluence, les utilisateurs mobiles peuvent recevoir de l'information adaptée à leur contexte géographique provenant d'un serveur Internet, tout en étant en mouvement.

ABSTRACT

The emergence of new communication and information technologies gives students access to a huge quantity of information at any time. Students' mobility, and their information sources are increasing. Although the low capacities of mobile devices and the low bandwidth of mobile networks, make the current paradigms not adapted to the mobile learning.

Mobile devices have certain constraints related to the size of their screen and keyboard, their computing speed, their consumption of energy, as well as to the diversity of their platform of execution. This implies difficulty to have access to specific information in a context of mobility. The next generation of mobile application intends to overcome all these limitations, while allowing users to have access to concise and adapted information, even when they are in motion.

In this thesis, we propose an innovative architecture of service allowing mobile students to receive data adapted to their geographical context, and to the characteristics of their mobile devices. From this architecture, we propose use cases that enable the detection of the user and the context of use, and also, support the access to adequate information by the students. Our model of service meets the majority of the constraints dictated by the use of mobile platforms. Detection, identification and transfer of data to mobile users are done using the Bluetooth protocol carried out on a J2ME platform. This platform is designed to work on several operating systems.

Our results showed that our model of application can be implemented on several mobile devices equipped with the execution platform Java. Moreover, when basic Bluetooth stations are placed at strategic places with high attendance, the mobile users can receive information adapted to their geographical context coming from an Internet server, while being in motion.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ.....	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES FIGURES	ix
CHAPITRE I INTRODUCTION	1
1.1 Définitions et concepts de base	2
1.2 Éléments de la problématique	3
1.3 Objectifs de recherche	5
1.4 Plan du mémoire	5
 CHAPITRE II MÉCANISMES D'ACCÈS À L'INFORMATION POUR DES APPAREILS MOBILES	 7
2.1 Contraintes liées à la diversité des plates-formes d'apprentissage	7
2.2 Requis de l'environnement d'apprentissage mobile	9
2.3 Approche de conception d'un mécanisme d'accès à l'information	10
2.3.1 Types d'apprentissage mobile.....	11
2.3.2 Les infrastructures des étudiants mobiles	11
2.3.3 Les protocoles mobiles.....	12
2.3.4 Les infrastructures des réseaux mobiles.....	13
2.4 Méthodologie d'analyse et de conception des applications	14
2.5 La notion de contexte	16
2.6 La gestion des documents dans un environnement d'apprentissage mobile	22
2.7 Implémentation des interfaces usager adaptables	23
2.7.1 Objets d'interaction abstraits versus objets d'interaction concrets	24
2.7.2 Description de XML	25
2.7.3 Exemples d'implémentation.....	26

CHAPITRE III MODÈLE ET ARCHITECTURE D'ACCÈS À	
L'ENVIRONNEMENT D'APPRENTISSAGE	29
3.1 Adaptabilité des applications mobiles d'apprentissage	30
3.1.1 Définition du problème	30
3.1.2 Analyse des requis du contexte d'exécution et d'utilisation	32
3.2 Architecture d'un modèle d'apprentissage mobile adaptable	34
3.2.1 Conceptualisation de l'environnement d'apprentissage mobile	34
3.2.2 Plate-forme de développement des applications d'apprentissage mobiles	41
3.3 Modèle d'application d'apprentissage mobile adaptable	43
3.3.1 Cas d'utilisation des applications d'apprentissage mobile adaptables.....	45
3.3.2 Modèle de configuration	48
3.3.3 Architecture du système de localisation des étudiants mobiles	52
 CHAPITRE IV IMPLÉMENTATION, MISE EN ŒUVRE ET RÉSULTATS	 57
4.1 Environnement d'implémentation	57
4.1.1 Environnement d'exécution matériel	57
4.1.2 Environnement d'exécution logiciel	58
4.2 Sous-ensemble de l'architecture implémentée et prototype	59
4.3 Analyse de performance	67
4.3.1 Choix des indices de performance	67
4.3.2 Réalisation des expériences.....	73
4.4 Comparaison et évaluation des modèles	84
 CHAPITRE V CONCLUSION ET DISCUSSION	 88
5.1 Synthèse des travaux	88
5.2 Limitations des travaux	89
5.3 Indications de recherches futures	90
 BIBLIOGRAPHIE	 91

LISTE DES FIGURES

Figure 2.1 Quelques appareils mobiles	9
Figure 2.2 Utilisation du système de gestion d'interface usager	15
Figure 2.3 Détecteur de contexte avec un Palm.....	18
Figure 2.4 Détecteur de contexte inséré dans un téléphone	20
Figure 2.5 Détecteur de contexte	20
Figure 2.6 Exemple de GUIs sur des plates-formes différentes.....	23
Figure 2.7 Adaptation des objets d'interface	25
Figure 2.8 Deux vues différentes générées automatiquement	26
Figure 3.1 Couches de l'application	44
Figure 3.2 Cas d'utilisation de notre application d'apprentissage mobile.....	46
Figure 3.3 Mode client/serveur	49
Figure 3.4 Exemple de connexion.....	50
Figure 3.5 Connexions du serveur	51
Figure 3.6 Téléchargement d'un fichier d'Internet.....	52
Figure 3.7 Adaptabilité géographique.....	53
Figure 3.8 Accès aux services sans fil	54
Figure 3.9 Communication multicast entre plusieurs étudiants.....	55
Figure 4.1 Diagramme de classe du prototype.....	60
Figure 4.2 Diagramme d'activité du prototype	61
Figure 4.3 Classe InputIdentification.....	62
Figure 4.4 Classe BluetoothDiscovery.....	63
Figure 4.5 Classe BluetoothConnection.....	65
Figure 4.6 Classe MessageCanvas	66
Figure 4.7 Classe UrlStreamConnection.....	66
Figure 4.8 Impact de la taille des fichiers sur le délai de transfert.....	68
Figure 4.9 Connexion d'un client mobile à plusieurs serveurs.....	70
Figure 4.10 Connexion de plusieurs clients à un serveur	71

Figure 4.11 Temps de réponse en fonction de la taille des fichiers	75
Figure 4.12 Écart type en fonction de la taille des fichiers (N=5)	76
Figure 4.13 Variation du délai de transfert entre un client et plusieurs serveurs.....	77
Figure 4.14 Variation du délai de transfert entre un client et plusieurs serveurs.....	79
Figure 4.15 Délai de transfert de plusieurs clients à un serveur	80
Figure 4.16 Identification des deux utilisateurs	81
Figure 4.17 Choix du mode connexion des utilisateurs	82
Figure 4.18 Recherche d'appareil mobile	83
Figure 4.19 Détection et envoi de fichier.....	83

CHAPITRE I

INTRODUCTION

La dernière décennie a vu le développement fulgurant d'appareils mobiles qui ne sont plus utilisés seulement par une élite. Les nouveaux téléphones cellulaires et assistants personnels sont introduits sur le marché à une cadence très élevée. Ces appareils mobiles, en plus de leurs fonctionnalités de base, offrent la possibilité d'accéder à des sites Internet, de télécharger des documents et de les organiser, et même d'envoyer des messages. De plus, l'information éducationnelle sur Internet croît exponentiellement et l'apprentissage basé sur le Web est un domaine de plus en plus important de recherche et développement. Cependant, les appareils mobiles, malgré leurs performances, ont des propriétés et des contraintes qui freinent leur utilisation dans le domaine de l'enseignement à distance. Des limitations telles qu'une capacité de traitement et une mémoire limitée, combinées à l'excès d'informations et de services ou au comportement "nomade" des étudiants font que les paradigmes d'apprentissage existants ne sont plus adaptés. L'objet de ce mémoire est donc de présenter une méthode de conception et d'implémentation d'un mécanisme d'accès à l'information dépendant du contexte pour des étudiants nomades. Dans ce mémoire, nous proposons une architecture de service novatrice en apprentissage mobile. Nous y traitons aussi de l'adaptabilité du contenu d'apprentissage mobile en fonction du contexte. Dans ce chapitre d'introduction, nous présentons d'abord quelques concepts de base qui permettront par la suite d'énoncer les éléments de la problématique et de préciser les objectifs de recherche. Un plan du mémoire complète ce chapitre.

1.1 Définitions et concepts de base

L'adaptabilité du contenu d'apprentissage destiné à des plates-formes mobiles par rapport au contexte d'utilisation constitue un des problèmes les plus intéressants de l'informatique mobile. Plusieurs recherches effectuées dans ce domaine proposent différents types de solutions. Parmi elles, figure l'utilisation d'agents mobiles qui pourraient détecter les spécificités du contexte d'utilisation et ainsi configurer le contenu d'apprentissage, afin qu'il puisse être présenté en tenant compte des avantages de son environnement d'exécution.

Un agent est une entité logicielle qui agit en faveur d'une autre entité (exemple: une personne, un autre agent); il est autonome et destiné à accomplir certaines tâches. Il réagit face aux événements externes provenant de son environnement. Selon Pierre (2003), un agent est un acteur artificiel qui remplit une ou plusieurs tâches d'assistance. C'est aussi une entité virtuelle capable d'agir sur son environnement, qui dispose d'une représentation partielle de cet environnement, qui peut communiquer avec d'autres agents dans un environnement multi-agent, et dont le comportement est la conséquence de ses observations, de ses connaissances et de ses interactions avec d'autres agents.

On peut comparer un agent à un objet informatique à la différence près qu'un agent possède en plus la notion de comportement qui lui donne une certaine autonomie. On compare souvent les agents et les objets par la citation suivante : "Objects do it for free, agents do it for money". Ainsi, lorsque l'on invoque la méthode d'un objet, ce dernier s'exécute sans se poser de questions, alors que l'agent va d'abord vérifier si l'exécution de la méthode n'est pas en contradiction avec ses propres plans. En d'autres termes, un agent a le contrôle de ses propres actions.

La disponibilité des services informatiques dans un environnement caractérisé par la mobilité des utilisateurs constitue une exigence de plus en plus importante pour les concepteurs d'applications basées sur Internet, dans le domaine des réseaux de communication. Selon Pierre (2003), la mobilité peut être définie comme la capacité d'accéder, à partir de n'importe quel endroit, à l'ensemble des services normalement disponibles dans un environnement fixe et câblé tels une maison et un bureau. De plus,

le concept d'informatique mobile réfère à la possibilité pour des usagers munis de périphériques portables ou d'ordinateurs mobiles d'accéder à des services et des applications évolués, à travers une infrastructure partagée de réseau, indépendamment ou par rapport à leur localisation physique ou à leur comportement.

On distingue essentiellement trois types de mobilité. La mobilité des unités pour laquelle le réseau offre aux terminaux ses services après leur identification et leur localisation. La mobilité personnelle qui concerne l'abonné qui est muni d'un numéro d'identification personnel et qui se déplace. Et la portabilité des services qui réfère à la capacité de rendre transparent du point de vue de l'utilisateur tout changement de terminal ou déplacement entre les nœuds d'un réseau. Ce qui garantirait l'accès aux mêmes services de n'importe quel lieu permettant l'accès au réseau.

L'avènement de la mobilité dans les réseaux de communications engendre une série de nouveaux services et d'applications dont les systèmes d'apprentissage électroniques (*e-learning*, *electronic learning*) qui permettent à un apprenant de travailler à un endroit donné, de se déconnecter, de changer d'endroit (en étant déconnecté), et de se reconnecter pour poursuivre son apprentissage. Le *m-learning* (*mobile electronic learning*) va plus loin et recherche la continuité des communications.

1.2 Éléments de la problématique

De nos jours, le nombre d'étudiants voulant avoir accès à de l'information d'apprentissage est en constante progression. Les ressources pédagogiques sont très limitées. La plus grande partie des étudiants cumulent travail et études, ce qui peut réduire leur temps potentiel d'accès aux documents pédagogiques. Les étudiants ont des rythmes d'apprentissage différents. Certains étudiants peuvent s'ennuyer dans une classe, alors que d'autres vont avoir des difficultés à suivre le rythme de la classe. Les cours actuellement donnés ne sont pas adaptés à cette situation. Ils ne prennent pas aussi en considération les progrès techniques et intègrent peu de contenu multimédia, ce qui les rend peu attrayants.

La mobilité des étudiants a rendu les appareils mobiles de plus en plus intéressants dans le domaine de l'apprentissage électronique mobile (*m-learning*). Mais la communication sans fil n'a pas encore atteint sa maturité et l'apprentissage basé sur des appareils électroniques sans fil est encore à l'état embryonnaire. Avec de plus en plus d'étudiants à temps partiel, l'accès à une base centralisée de connaissances est devenu essentiel et les appareils mobiles sans fil pourraient rendre l'accès à l'information possible de partout virtuellement, d'où la portabilité et l'ubiquité des services d'apprentissage. De plus, un appareil mobile sans fil permettrait d'accroître les performances des étudiants en leur offrant la possibilité d'étudier à tout moment du jour ou de la nuit et de pouvoir accéder à de l'information, dépendant du lieu d'apprentissage.

Cependant, le *m-learning* se heurte à beaucoup de contraintes. Un des plus grands défis pour les concepteurs de contenu d'apprentissage basé sur des appareils mobiles réside dans la diversité des plates-formes qui ont pour la plupart des spécifications et des contraintes propres. De plus, les utilisateurs d'appareils sans fil sont très mobiles et les applications d'apprentissage doivent pouvoir garantir l'accès à de l'information là ou il le faut, quand il le faut, et à un coût raisonnable.

Une solution à la contrainte d'hétérogénéité des plates-formes d'exécution des applications mobiles est la séparation entre l'interface usager et le reste de l'application. Ce qui permet de faire des changements au niveau de l'interface usager sans pour autant affecter l'implémentation du reste de l'application. On aurait ainsi une interface usager pour chaque plate-forme. Cependant, ceci pose aussi des problèmes, en l'occurrence la répétition inutile de l'implémentation pour chaque plate-forme. Dans ce cas, les révisions de la conception seraient effectuées plusieurs fois et l'ajout d'une nouvelle plate-forme se traduirait par une ré-implémentation.

La solution la plus souvent préconisée pour résoudre le changement de contexte est l'utilisation de capteurs sur les appareils mobiles. Ces senseurs permettent ainsi aux applications mobiles de détecter le lieu et les conditions d'utilisation et d'adapter leurs fonctionnalités. Cependant, cette approche pose aussi des problèmes, en l'occurrence la

standardisation des senseurs utilisés ainsi que leur connectivité aux appareils mobiles. Les applications d'apprentissage doivent pouvoir être exécutées sur des appareils mobiles standard afin de pouvoir être utilisées par le plus d'étudiants possibles. D'où la recherche sur l'adaptabilité du contenu d'enseignement à distance en contexte de mobilité. Le contenu d'apprentissage basé sur des appareils mobiles doit s'adapter aux contraintes de communication, de consommation d'énergie, de puissance de calcul et d'interface graphique de la plate-forme d'exécution.

1.3 Objectifs de recherche

L'objectif principal de ce mémoire est de concevoir des mécanismes d'accès au contenu pédagogique basé sur des appareils mobiles en fonction du contexte d'apprentissage des étudiants. De manière plus spécifique, nous visons les objectifs suivants :

- définir les requis d'apprentissage dans un contexte de mobilité ;
- analyser différentes technologies en faisant ressortir leurs avantages pour une nouvelle architecture d'apprentissage mobile ;
- concevoir une architecture de service novatrice permettant aux étudiants nomades d'avoir accès à un contenu d'enseignement en fonction de leur lieu d'apprentissage ;
- évaluer les performances techniques et conceptuelles de l'architecture proposée en vue de faire ressortir ses avantages et ses inconvénients, afin de la comparer aux solutions existantes.

1.4 Plan du mémoire

Outre l'introduction, ce mémoire comprend essentiellement quatre chapitres. Le chapitre 2 dresse l'état de l'art dans le domaine de l'adaptabilité du contenu des appareils mobiles. Il présente les tendances et les derniers développements effectués dans ce domaine par différentes équipes de recherche. Le modèle conceptuel et la méthodologie adaptée pour l'accès au contenu d'enseignement en fonction du contexte d'apprentissage des étudiants nomades sont présentés de façon détaillée dans le chapitre

3. Ce chapitre justifie également les requis et les différents choix technologiques effectués. Dans le chapitre 4, le modèle conceptuel est implémenté et une analyse de performance nous permet d'évaluer sa viabilité. Enfin, le chapitre 5, en guise de conclusion, effectue une synthèse, situe la contribution de nos travaux dans le domaine et esquisse quelques indications de recherches futures en vue de combler les inévitables lacunes de réalisation.

CHAPITRE II

MÉCANISMES D'ACCÈS À L'INFORMATION POUR DES APPAREILS MOBILES

Les technologies sans fil deviennent de plus en plus étendues. Les professeurs et étudiants sont de plus en plus mobiles et utilisent de plus en plus des appareils sans fil, interagissant ainsi avec des environnements d'apprentissage utilisant des technologies de pointe. Le regain de mobilité permis par les appareils sans fil pousse donc les applications d'apprentissage mobile à devoir s'adapter aux environnements d'apprentissage qui deviennent dynamiques. La capacité et le coût d'accès aux réseaux sans fil ainsi que la disponibilité des appareils mobiles apportent des changements au niveau du temps d'utilisation de ces appareils qui devient plus long et au niveau de l'espace où se trouve l'utilisateur qui peut changer pendant l'exécution d'une tâche. Une instanciation des paramètres d'apprentissage dynamique est appelée contexte d'utilisation.

Le présent chapitre décrit brièvement des mécanismes d'accès à l'information pour des appareils mobiles. Nous présentons les requis et les contraintes liés à l'élaboration d'un environnement d'apprentissage adaptable basé sur les appareils mobiles. Nous décrirons ensuite le concept de contexte et présenterons des solutions potentielles permettant l'accès à de l'information contextuelle, et la collaboration avec d'autres apprenants. Après évaluation de ces solutions, nous mettons en évidence les nombreuses lacunes qui justifient l'émergence de solutions alternatives.

2.1 Contraintes liées à la diversité des plates-formes d'apprentissage

À cause de la diversité de leurs caractéristiques, les ordinateurs portables (ou de poche) ont causé une très forte poussée dans la recherche sur l'interopérabilité de leurs

services. Et dans le même ordre d'idée, l'adaptation du contenu d'apprentissage par rapport au contexte dans lequel se trouvent les étudiants représente un grand défi.

L'accès aux données provenant des appareils mobiles suppose la prise en considération de certaines contraintes qui n'existent pas pour les stations de travail fixes. Plusieurs restrictions influencent la manière dont l'information peut être présentée sur les appareils mobiles (des téléphones, des ordinateurs de poche ou des PDAs). Chacune de ces restrictions doit être prise en compte lors de la conception du contenu qui doit être envoyé et affiché. Parmi les contraintes les plus importantes, nous pouvons noter (Gaedke, Beigl, Gellersen et Segor, 1998):

- La consommation d'énergie : une attention spéciale doit être portée afin que les applications ne soient pas énergivores. Par exemple, une utilisation excessive de périphériques de sortie tels l'audio devrait être évitée dans les applications qui s'exécutent à partir des appareils mobiles.
- La puissance de calcul : les appareils mobiles ont, en général, beaucoup moins de puissance de calcul que les stations de travail. De ce fait, les contenus qui requièrent beaucoup de calcul telle la compression vidéo ne peuvent pas être utilisés.
- La capacité d'affichage : les PDA et les autres types d'ordinateurs mobiles ont une surface d'affichage, qui est beaucoup plus petite que les stations de travail et qui varie énormément d'un appareil à l'autre.
- La communication : l'information à envoyer aux appareils mobiles doit être adaptée à la bande passante disponible et aux coûts de transmission. De plus l'hétérogénéité des appareils mobiles et de leurs moyens de communication rend d'autant plus difficile la standardisation de la communication entre appareils mobiles.

Les contraintes énumérées précédemment et qui s'appliquent au transfert de données vers des appareils mobiles démontrent le requis d'adaptabilité à plus d'une plate-forme. Par exemple, le contenu d'une page Web ou le document que s'échangent

deux appareils mobiles doivent être adaptés en prenant en considération le choix du média, la disposition des pages et le volume de données à transférer en considérant la puissance de calcul, la capacité d’affichage et la bande passante disponible au niveau des appareils.

2.2 Requis de l’environnement d’apprentissage mobile

La diversité croissante des terminaux informatiques comme illustré à la Figure 2.1 constitue un défi de taille pour les concepteurs informatiques (Eisenstein et al., 2000) :

- les applications interactives doivent pouvoir s’exécuter sur plusieurs plates-formes différentes, allant de la station de travail fixe au simple téléphone cellulaire;
- les applications destinées à plusieurs plates-formes doivent s’adapter aux périphériques d’entrée et de sortie des plates-formes des appareils mobiles;
- les applications doivent être conçues afin de s’adapter et de prendre avantage des différents contextes de leur utilisation;
- les applications mobiles doivent s’adapter au changement du contexte d’utilisation ou de l’environnement, même pendant l’exécution d’une fonction.

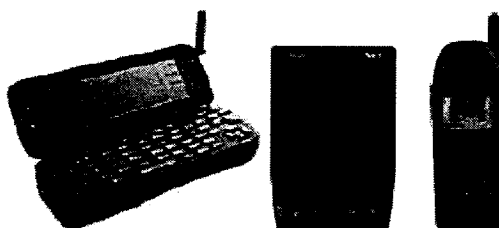


Figure 2.1 Quelques appareils mobiles

De plus, la problématique devient plus complexe, si on prend en considération le fait que l’usager peut bouger et utiliser plusieurs plates-formes pendant l’exécution

d'une ou de plusieurs tâches. Un lecteur qui veut acheter un livre pourrait d'abord le chercher de sa station de travail, sauvegarder les références des livres qu'il a trouvées sur un aide mémoire électronique (PDA : Personal Digital Assistant) afin de les lire dans le métro et ensuite commander les livres à partir de son téléphone cellulaire. Plusieurs utilisateurs pourraient aussi vouloir collaborer avec des appareils mobiles ayant des plates-formes hétérogènes. Par exemple, un étudiant et son professeur pourraient vouloir confronter des notes, l'un sur une station de travail et l'autre sur un PDA. En somme, les applications doivent être sensibles aux :

- Plates-formes : en ayant une présentation qui s'adapte à la surface d'écran utilisable, aux couleurs disponibles, à la résolution de l'écran, à la largeur de bande et aux types de médias de communication disponibles sur l'appareil.
- Interactions : en ayant des mécanismes de mémorisation de la taille des fenêtres et de leur emplacement ainsi que des préférences des utilisateurs.
- Utilisateurs : en s'adaptant au niveau d'expérience des utilisateurs, à leurs aptitudes, à leurs convenances et à leurs préférences.

2.3 Approche de conception d'un mécanisme d'accès à l'information

L'apprentissage mobile se trouve à l'intersection de l'apprentissage électronique et de l'informatique mobile. Afin de simplifier l'étude de ce domaine, plusieurs environnements ont été proposés dont celui présenté par LEUNG et CHAN (2003) où sont présentés quatre niveaux qui sont les applications d'apprentissage mobiles, les infrastructures mobiles, les protocoles mobiles et les réseaux mobiles.

L'environnement pour apprentissage mobile définit plusieurs niveaux de fonctionnalités, simplifiant ainsi la conception et le développement afin que les différents acteurs du domaine (vendeurs, fournisseurs, concepteurs) puissent adresser des niveaux individuels. En utilisant cet environnement, les fonctionnalités des applications mobiles sont distribuées parmi les différents niveaux. Chaque niveau offre des services au niveau qui lui est supérieur. Dans le niveau des applications d'apprentissage mobile, plusieurs types d'applications incluant des fonctionnalités

nouvelles deviennent possibles, et plusieurs applications électroniques peuvent être appliquées à l'environnement mobile. Dans le niveau des infrastructures mobiles, la conception de nouvelles applications d'apprentissage mobile devrait prendre en compte les nouvelles capacités des appareils mobiles. Dans le niveau des réseaux mobiles, la qualité de service dépend des ressources réseaux et de leurs capacités.

2.3.1 Types d'apprentissage mobile

Il existe plusieurs moyens permettant l'accès à de l'information pédagogique en dehors des salles de classe. Par exemple, les étudiants pourraient visiter des musées pour observer certaines pièces ou collecter certains spécimens lors d'une étude sur le terrain. Nous pourrions donc penser à un paradigme qui permettrait d'envoyer de l'information complémentaire pour chaque pièce du musée à chaque étudiant dès qu'ils sont à proximité de la pièce. Cette information pourrait être sauvegardée et ensuite envoyée à un ordinateur sur lequel l'étudiant pourrait faire le rapport de sa visite. Nous pouvons aussi avoir un professeur qui prendrait des photos de spécimens afin d'en faire un support de travail en envoyant cette photo à tous ses étudiants. La gestion des activités d'apprentissage mobile peut gérer la position d'un étudiant et évaluer le besoin de cet étudiant dans une situation particulière, de manière à ce qu'il soit possible d'organiser son cours conséquemment, améliorant ainsi l'efficacité de l'étudiant. Nous pouvons penser aussi à un paradigme permettant la collaboration entre étudiants grâce au partage de documents parmi des membres d'un seul cours ou avec un professeur en fonction d'une situation géographique d'apprentissage.

2.3.2 Les infrastructures des étudiants mobiles

Les infrastructures et les réseaux sans fil peuvent fournir du texte, du son, de la vidéo sur demande et des services d'information. Les étudiants auraient besoin de technologies telles des appareils mobiles appropriés, ainsi que des réseaux sans fil à large bande. La connectivité est importante parce que la déconnexion affecte sérieusement la qualité des services d'information. De plus les coûts des

communications ne devraient pas être élevés étant donné que les étudiants ne disposent pas d'un gros budget. Le type d'appareils mobiles pouvant être utilisé dans un contexte d'apprentissage mobile est très varié. La plupart des téléphones cellulaires existant sur le marché sont dotés d'appareil photo et de technologie de communication tel le Bluetooth qui permet l'interconnexion automatique de différents utilisateurs ainsi que l'échange de données. Les Assistants Numériques Personnels (PDA) avec modem sans fil ou des appareils avec des fonctionnalités logicielles appropriées peuvent être utilisés. Ces appareils sont conçus aussi bien pour la communication que pour des opérations requérant des calculs. Dans un avenir proche, ces fonctionnalités pourront être combinées dans des appareils mobiles intelligents. Un système d'exploitation est aussi requis et il en existe qui peuvent s'adapter aux capacités des appareils mobiles tels Symbian OS[®] ou Windows CE[®].

2.3.3 Les protocoles mobiles

Les protocoles mobiles permettent de connecter plusieurs applications, plusieurs outils, plusieurs réseaux et plusieurs technologies. Arriver à une abstraction entre le niveau matériel (appareils mobiles) et les technologies sans fil est important. La couche protocole permet l'interconnexion de différents étudiants mobiles avec plusieurs réseaux mobiles et différents appareils. Cette couche permet d'ajuster les fluctuations de bande passante et de délais, ainsi que les changements de lieu pour les étudiants nomades. Elle peut donner aux utilisateurs d'applications un meilleur temps de réponse et une meilleure fiabilité. Les techniques d'optimisation typiques au niveau des protocoles mobiles incluent la compression d'en-tête, la réponse retardée ou "delayed acknowledgment" et la concaténation de plusieurs petits paquets en un seul afin de réduire le trafic des réseaux sans fil.

Plusieurs standards et protocoles sont en mesure de supporter le développement des applications d'apprentissage mobiles à travers différentes plate-formes, et de permettre que ces applications puissent communiquer et s'envoyer des données. Le WAP utilise un large éventail d'applications et de protocoles différents et il peut faciliter

l'interopérabilité entre différents réseaux sans fil, différents appareils et applications. Un mini fureteur pris comme logiciel client peut supporter aussi bien du texte, des images que du contenu Internet standard. De plus, dans un futur proche, le contenu web pourra être accessible à partir des PC, TV, PDA et des téléphones cellulaires. Il existe des protocoles mobiles qui transitent par les fournisseurs de service de téléphonie et des protocoles qui permettent la communication entre appareils mobiles et l'échange direct d'information entre utilisateurs.

2.3.4 Les infrastructures des réseaux mobiles

En plus des protocoles et des appareils mobiles, les mécanismes d'apprentissage mobile dépendent du support réseau. Le débit de transmission et le taux de couverture sont les deux facteurs les plus importants. Parce que plusieurs applications d'apprentissage mobile requièrent des petits messages textes, ils pourraient utiliser les SMS du réseau européen GSM ou d'autres réseaux. Les développeurs d'applications sont en train de standardiser la troisième génération de réseaux sans fil (UMTS) qui peuvent supporter une largeur de bande de l'ordre du Mégabits par seconde. Un autre facteur important est le support du multicast par lequel une discussion avec un groupe d'étudiants est possible. En général l'apprentissage mobile requiert un service sans fil de qualité qui affecte la performance des applications d'apprentissage sans fil en termes de délai, de perte et d'autres attributs de qualité. La gestion de la localisation est aussi un requis et fait partie intégrante de plusieurs applications d'apprentissage, spécialement ceux impliquant la recherche de lieux ou d'étudiants. Un protocole de positionnement mobile (MPP), actuellement dans la version 1.1, a été développé par Ericsson. Il s'agit d'un protocole basé sur Internet et employé dans des applications qui dépendent de leur position géographique pour se connecter à un serveur de positionnement mobile. Par ce protocole, il est possible de demander la position de terminaux mobiles.

En outre, des réseaux sans fil fiables sont nécessaires. Les étudiants peuvent accéder au contenu d'apprentissage mobile même avec un réseau n'assurant pas une certaine qualité de service. Finalement, le "roaming" à travers plusieurs réseaux

hétérogènes peut permettre aux étudiants d'accéder aux applications d'apprentissage mobile de partout, même quand la couverture d'un seul réseau est instable. Parmi les protocoles de communication qui existent au niveau des réseaux mobiles, nous avons le Bluetooth et le Wi-Fi. La technologie Bluetooth est une technologie de réseau personnel sans fil (noté WPAN pour Wireless Personal Area Network), c'est-à-dire une technologie de réseaux sans fil d'une faible portée, de l'ordre de quelques dizaines de mètres à un peu moins d'une centaine de mètres, permettant de relier des périphériques (imprimantes, téléphones portables, appareils domestiques, oreillettes sans fil, etc.) et des ordinateurs et assistants personnels (PDA) entre eux sans liaison filaire. La norme IEEE 802.11 (ISO/IEC 8802-11) appelée Wi-Fi (Wireless Fidelity) est un standard international décrivant les caractéristiques d'un réseau local sans fil (WLAN). Grâce au Wi-Fi, il est possible de créer des réseaux locaux sans fil à haut débit pour peu que la station à connecter ne soit pas trop distante par rapport au point d'accès. Dans la pratique, le Wi-Fi permet de relier des ordinateurs portables, des ordinateurs de bureau, des assistants personnels (PDA) ou tout type de périphérique notamment des téléphones portables à une liaison haut débit (11 Mbps ou supérieur) sur un rayon de plusieurs dizaines de mètres en intérieur (généralement entre une vingtaine et une cinquantaine de mètres) à plusieurs centaines de mètres en environnement ouvert.

2.4 Méthodologie d'analyse et de conception des applications

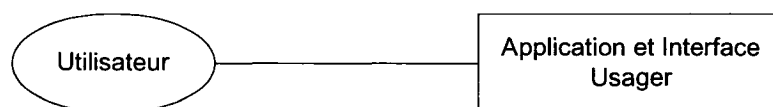
Un problème de base des applications destinées à plusieurs plates-formes ou s'exécutant sur un serveur consiste à pouvoir fournir une panoplie d'accès et de mécanismes d'interaction qui donnent les mêmes réponses et les mêmes fonctionnalités, indépendamment de la plate-forme. C'est ainsi que Jacob (2001) propose de diviser la conception des applications mobiles en niveaux sémantique, syntaxique et lexical :

- le niveau sémantique décrit les fonctions exécutées par le système. Cela correspond aux requis fonctionnels de l'application. Il s'occupe aussi de la gestion des erreurs ;

- le niveau syntaxique décrit les séquences d'entrée et de sortie nécessaires pour l'exécution des fonctionnalités. La conception du niveau syntaxique englobe l'élaboration des séquences d'entrée-sortie logiques et des opérations sémantiques ;
- le niveau lexical décrit la relation entre les opérations d'entrée-sortie et le matériel.

Le concept de la séparation entre l'interface usager et le reste de l'application est décrit comme un principe clé. En effet, cela permet de faire des changements au niveau de l'interface usager sans pour autant que cela n'affecte le code du reste de l'application. Il peut donc y avoir plusieurs interfaces usager pour la même application. De plus, en utilisant un modèle à plusieurs niveaux, il est possible de séparer le code du niveau sémantique de celui du niveau syntaxique et du niveau lexical, comme montré à la Figure 2.2. D'où le concept de Système de Gestion d'Interface Usager (UIMS : User Interface Management System) (Hix, 1990) qui est un composant logiciel gérant l'interface usager et qui est séparé du reste de l'application.

Avant



Après

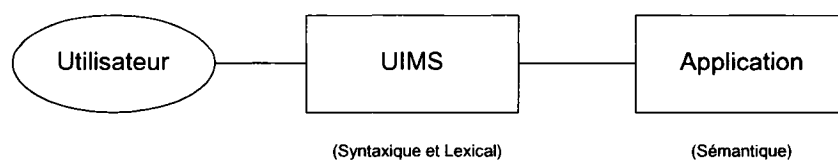


Figure 2.2 Utilisation du système de gestion d'interface usager

En prenant en considération cette philosophie de conception d'interface usager, nous pourrions penser à une application ayant une interface usager interchangeable avec des fonctionnalités qui ne changent pas. Ceci pourrait aboutir en une interface usager qui s'adapterait à différents contextes ou situations (ex : utilisateur en classe, couché à la maison, en session de travail, etc..). Ainsi, lors de la mise en marche du système, il détecterait automatiquement l'état de l'utilisateur, ainsi que les périphériques d'entrée-sortie disponibles et afficherait une interface usager qui soit adaptée.

2.5 La notion de contexte

Le temps disponible pour configurer les applications dans des appareils ultra mobiles peut être très petit. De plus, le fait que ces appareils soient utilisés pendant l'exécution d'une tâche ou servent d'outils pour exécuter une tâche implique que le temps d'interaction entre l'utilisation et l'application s'en trouve réduit. Cela a conduit au passage vers une interaction homme-machine implicite. Plusieurs recherches ont été effectuées dans le "context-awareness" (prise en compte de l'environnement dans lequel l'application se trouve). Dans un domaine général, le contexte se définit comme l'ensemble des circonstances dans lesquelles s'insère un fait. Ce terme est utilisé dans le domaine informatique pour la description d'un environnement, de la situation, de l'état, des interfaces, des tâches, etc. Il a été montré que la détection du contexte est utile à différents niveaux pour un dispositif mobile. Au niveau système, le contexte pourrait être utilisé pour la détection des ressources disponibles et pour la gestion du courant utilisé. Au niveau application, il pourrait être utilisé pour l'adaptation des applications et des services qui dépendent de contextes particuliers. Au niveau des interfaces usager, l'utilisation du concept de contexte facilite la transition des interactions qui, au lieu d'être explicites, sont implicites.

Un système d'apprentissage qui examine le contexte d'apprentissage adapte le processus d'apprentissage en respectant les changements de contexte. Bien que ceci ne soit pas une nouvelle idée, la détection et la prise en compte du contexte est de plus en plus essentielle dans l'apprentissage mobile du fait que le contexte d'apprentissage

devient plus dynamique et plus complexe. Cependant, la conscience du contexte n'est pas facile à implémenter. La diversité des technologies mobiles et sans fil, et la nature dynamique des environnements mobiles rend compliqué la conscience du contexte. Par conséquent, les informations contextuelles comme les capacités d'affichage des appareils mobiles, les périphériques de communication et de transfert d'information sont d'habitude prédéterminés et encodés dans les logiciels des utilisateurs.

Les applications d'apprentissage conscientes de leur contexte d'utilisation mettent plus l'emphasis sur les étudiants mobiles qui ont en leur possession des appareils portables connectés, tels les PDA munis de senseurs comme des récepteurs GPS, des réseaux de données sans fil, des caméras, etc., et des senseurs logiciels tels que les gestionnaires de congestion réseau, les analyseurs de notations d'enchaînement, des analyseurs de comportement d'étudiant. Ces senseurs peuvent détecter la location, l'activité, la congestion réseau, etc. Les applications d'apprentissage mobile réagissent aux informations recueillies par les détecteurs de contexte en modifiant leur environnement d'apprentissage. Le matériel de cours présenté aux étudiants prend donc en considération l'environnement d'apprentissage et fournit des informations plus appropriées afin de prendre en compte leurs besoins, dépendant du contexte dans lequel ils se trouvent.

Plusieurs personnes se sont penchées sur le développement de mécanismes permettant la prise en compte du contexte d'utilisation. Afin de communiquer aux applications mobiles un changement de contexte ou tout simplement de configurer certains de leurs aspects visuels, les méthodes conventionnelles (clavier, souris, etc.) se sont avérées très peu pratiques. Il a donc fallu donner aux applications leurs propres moyens de se configurer sans pour autant qu'il y ait une participation active de l'utilisateur. Il a donc été développé un dispositif de reconnaissance de contexte (Schmidt, 2000) constitué :

- d'un photomètre ;
- d'un senseur d'accélération ;
- d'un senseur d'infrarouge passif ;

- d'une sonde de contact ;
- d'un thermomètre.

Ce dispositif montré à la Figure 2.3, est contrôlé par un microcontrôleur BASICTiger qui lie les différents senseurs et envoie les données au PDA par une connexion série à 19 200 bits par secondes. À chaque fois que l'application désire une mise à jour, elle envoie une commande "GET" au dispositif de reconnaissance de contexte qui lui répond par un message contenant l'état encodé du contexte. Ce genre de dispositif peut s'avérer très utile dans des applications aussi bien pour l'entrée des données que pour la sortie des résultats.

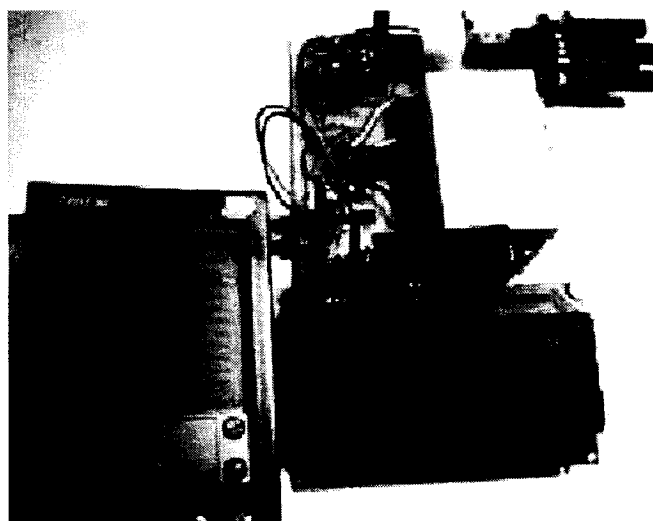


Figure 2.3 Détecteur de contexte avec un Palm

En ce qui concerne l'entrée des données, ce dispositif permet tout d'abord d'adapter la procédure d'entrée des données à la situation (filtre audio, algorithme de reconnaissance, etc.). Il permet ensuite de réduire le temps requis pour configurer les dispositifs mobiles et enfin, il permet de n'offrir que les menus propres à des contextes particuliers, réduisant ainsi l'espace requis pour la sélection des menus au niveau de l'application.

Pour la présentation des résultats, les informations liées au contexte permettent d'adapter la sortie des applications à la situation et permettent de réduire les interruptions de l'utilisateur dans sa tâche pour configurer l'application. Le concept de conscience du contexte a été implémenté et a abouti à une application NotePad sur PalmPilot qui s'allume dès qu'on le tient entre les mains et qui s'éteint quand on le dépose. La taille de la police des caractères s'agrandit quand l'appareil bouge et revient à sa taille normale quand l'appareil est en position stable. La luminosité de l'écran change selon la luminosité ambiante. Lorsque l'utilisateur n'est pas seul et qu'il n'est pas en train d'écrire, l'écran de l'appareil est caché par une image.

Le même concept a été utilisé par Gellersen et al. (2002), plus spécifiquement dans le projet TEA (Technology for Enabling Awareness) où plusieurs senseurs assez simples ont été utilisés. En l'occurrence, deux photomètres, deux microphones, un accéléromètre à deux axes, un senseur sensible à la conductance de la peau et un thermomètre. Ces senseurs sont lus par un microcontrôleur (PIC16F877) qui communique avec un appareil mobile à travers un port série. Ces senseurs captent individuellement un petit aspect de leur environnement. Toutes ces informations mises en commun aboutissent ainsi à une vision d'ensemble des caractéristiques de l'environnement. L'utilisation de plusieurs petits senseurs a pour objectif de disposer d'information requérant un minimum de traitement. De ce fait, la puissance de calcul de l'appareil est plus sollicitée par exemple, pour le traitement du son que pour l'acquisition de celui-ci. L'architecture du TEA a été bâtie sous forme de couches pour le traitement du contexte basé sur des capteurs. Nous avons dans la couche la plus basse, le flux de données provenant des capteurs. La couche suivante s'occupe d'extraire les caractéristiques importantes de ces données. Et enfin, la dernière couche permet d'avoir une vue d'ensemble du contexte. Le prototype qui a été développé est indépendant de la plate-forme avec laquelle il interagit. En effet, il pourrait être utilisé avec un Palm, un ordinateur symbiotique ou un téléphone portable avec lesquels il pourrait communiquer à travers leur interface série.

L'implémentation de ce concept a été réalisée avec un téléphone mobile où les utilisateurs peuvent définir des profils pour différentes situations et peuvent spécifier le comportement désiré pour ces situations comme le montre les Figures 2.4 et 2.5. La technologie WAP (Wireless Application Protocol) a été utilisée pour supporter la configuration de l'appareil.

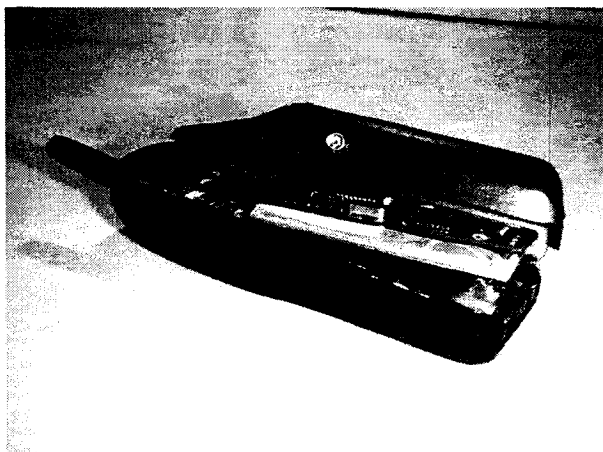


Figure 2.4 Détecteur de contexte inséré dans un téléphone

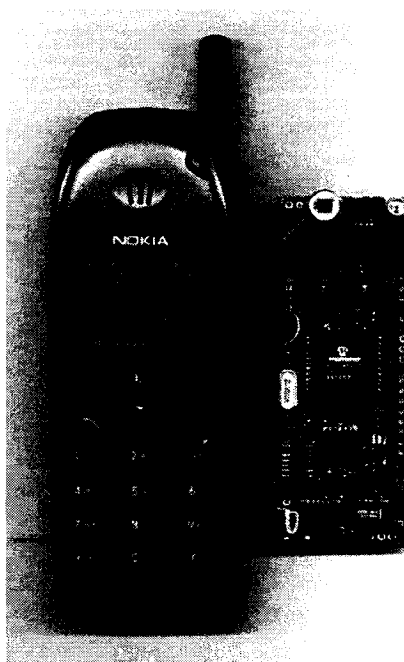


Figure 2.5 Détecteur de contexte

Dans les expériences qui ont été réalisées avec le TEA (Van Laerhoven, 2000), les situations « en main », « sur table », « dans une pochette » et « à l'extérieur » ont été retenues avec 87% de réussite (ou plus dépendamment du contexte) avec un délai de 30 secondes.

En somme, cette expérience a permis de trouver une architecture à plusieurs niveaux permettant l'estimation du contexte d'utilisation. De plus, des senseurs tels les microphones, les senseurs d'accélération et les photomètres peuvent facilement être utilisables pour plusieurs types d'applications. Cependant la perception de l'environnement aurait pu être améliorée en utilisant plusieurs capteurs de même type, par exemple, plusieurs photomètres avec des orientations différentes ou plusieurs microphones. Ces senseurs pourraient aussi être utilisés dans un environnement d'apprentissage facilitant ainsi la configuration des applications d'apprentissage. Ainsi la taille de police de l'information d'apprentissage pourrait être adaptée en fonction de la mobilité de l'utilisateur par exemple.

Nous avons vu dans cette partie le développement d'applications conscientes de leur contexte d'utilisation. Nous pourrions penser appliquer ces mêmes méthodes au domaine de l'apprentissage mobile où le contexte se produit du fait des interactions entre les étudiants et les professeurs ou celles entre des étudiants et des ordinateurs. Dans un environnement d'apprentissage, les applications adaptables doivent prendre en considération une variété de variables contextuelles parmi lesquelles nous pouvons citer : l'adaptation aux objectifs pédagogiques sous-jacents à la situation d'apprentissage, à la nature et aux besoins des apprenants impliqués, au niveau de détail requis pour atteindre les objectifs pédagogiques et aux contraintes d'ordre technique qui forment le contexte d'apprentissage. Un contexte d'apprentissage est dynamique, interactif et situé, mais un contexte peut mener à des formes d'interaction plus variées. Les différents exemples cités ci-dessus seraient difficilement applicables dans le domaine de l'apprentissage mobile. En effet, ceux-ci utilisent des senseurs qui sont rajoutés aux appareils mobiles et qui forceraient les étudiants à acheter du matériel supplémentaire afin d'utiliser leur téléphone dans un contexte d'apprentissage.

2.6 La gestion des documents dans un environnement d'apprentissage mobile

Afin de décrire la notion de gestion de documents dans le contexte d'applications liées à des appareils mobiles, prenons comme exemple la communauté estudiantine où l'usage des appareils mobiles est de plus en plus populaire. Il paraît donc intéressant de permettre à ces étudiants de pouvoir utiliser ces outils dans le cadre de leur apprentissage. Cependant, certaines contraintes existent quant à la présentation des ces informations qui peuvent être de nature hétéroclite. Un aspect important de l'apprentissage est le besoin de rassembler et de combiner les documents de cours provenant de plusieurs sources (et de nature différente) et d'avoir à discuter et à partager ces informations avec des coéquipiers.

Dans le projet DIANE (Klein et König-Ries, 2002), il est fait état d'un concept d'interface usager graphique qui permettrait, sans disposer des documents, d'obtenir de l'information sur leurs contenus, leurs types et leurs formats. Il serait possible de voir cette information à plusieurs niveaux de détails et de grouper les documents par rapport à un sujet spécifique. De plus, l'utilisateur pourrait même voir si les documents sont sur son appareil ou sur un serveur quelconque et décider de les télécharger. Un dernier requis serait que l'interface devrait être adaptable à des appareils mobiles différents et qui ont des capacités graphiques différentes. Un concept de GUI (Graphical User Interface) qui répond à ces requis est montré à la Figure 2.6. La base de cette approche est formée par deux ontologies. Une ontologie se définit comme un ensemble de concepts qui forment un réseau de relations pour décrire le contenu sémantique de ces concepts.

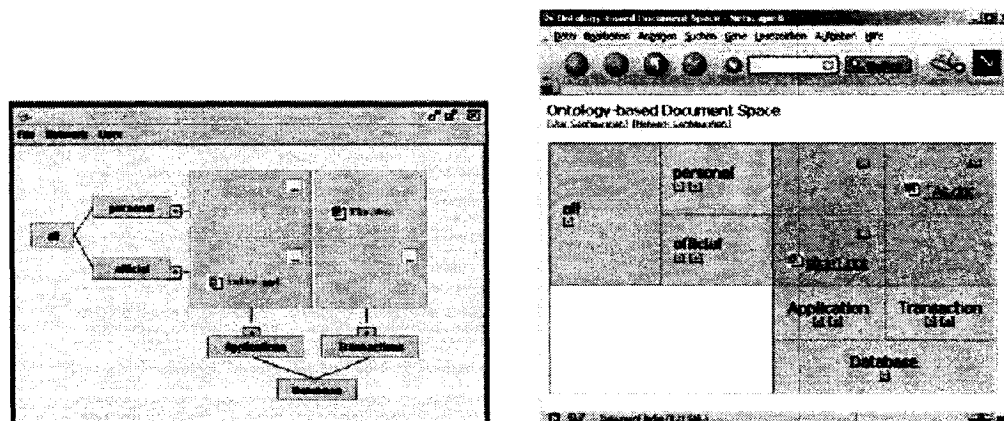


Figure 2.6 Exemple de GUIs sur des plates-formes différentes

La relation *isSubclassOf* (« est Sous-classe De ») est utilisée pour décrire la sémantique *is a* (« est une »). Les différents concepts sont organisés dans une arborescence avec une racine qui représente le terme le plus général et qui couvre les autres concepts de l'ontologie. Ici la première ontologie utilisée, sur l'axe vertical, décrit la source des documents et la seconde représente les différents domaines auxquels peuvent être associés les documents. L'espace entre les deux axes est réservé aux documents. Selon le degré de détail voulu par l'utilisateur, les concepts des deux ontologies peuvent être rétractés ou déployés. L'utilisateur peut aussi commencer par une vue d'ensemble du domaine et ensuite voir des thèmes plus spécifiques. Ce GUI ne fournit pas seulement une vue d'ensemble de la disponibilité des documents mais peut aussi servir à des fins d'interactions entre les usagers. Pour chercher un document spécifique, l'usager pourrait marquer une position spécifique dans le diagramme ou même sélectionner plusieurs concepts.

2.7 Implémentation des interfaces usager adaptables

Plusieurs méthodes permettant l'adaptation des interfaces usagers des applications mobiles sont proposées dans la littérature. Ces méthodes sont utilisées dans la phase de conception ou d'implémentation de ces applications.

2.7.1 Objets d'interaction abstraits versus objets d'interaction concrets

Un Objet d'Interaction Abstrait (AIO) aussi appelé “widget” est tout élément qui permet aux utilisateurs d'une application de visualiser ou de manipuler de l'information ou d'accomplir une tâche interactive. Du point de vue modélisation, ces objets sont souvent considérés comme les éléments atomiques de la constitution d'une application d'apprentissage adaptable. Nous pouvons faire une distinction entre les AIO et les Objets d'Interaction Concrets (CIO). Les CIO sont des objets d'interface qui sont exécutables sur une plate-forme sans traitement additionnel nécessaire. Bien que les CIO soient exécutables, il n'est pas nécessaire de comprendre leurs détails d'implémentation et leur comportement ne dépend pas du reste de l'application, à moins qu'il y ait un besoin de paramétrisation. À la différence d'un CIO, un AIO n'est exécutable sur aucune plate-forme et fait abstraction des caractéristiques de celles-ci. Pour cette raison les AIO sont complètement portables.

Ainsi Eisenstein, Vanderdonckt et Puerta (2000), nous montrent une architecture du modèle de présentation; chaque AIO est le parent de plusieurs CIOs dont chacun hérite des propriétés de son AIO, tout en ayant des caractéristiques qui lui sont propres et qui dépendent de la plate-forme à laquelle il est dédié. Nous pouvons voir à la Figure 2.7 un exemple de ce concept où un bouton-poussoir a plusieurs implémentations dépendantes de la plate-forme. Ainsi il peut être représenté sous la forme d'un bouton “ActiveX” sur une plate-forme d'ordinateur de bureau, comme un bouton PDA qui est propre aux PDA ou comme un bouton XML pour un téléphone cellulaire.

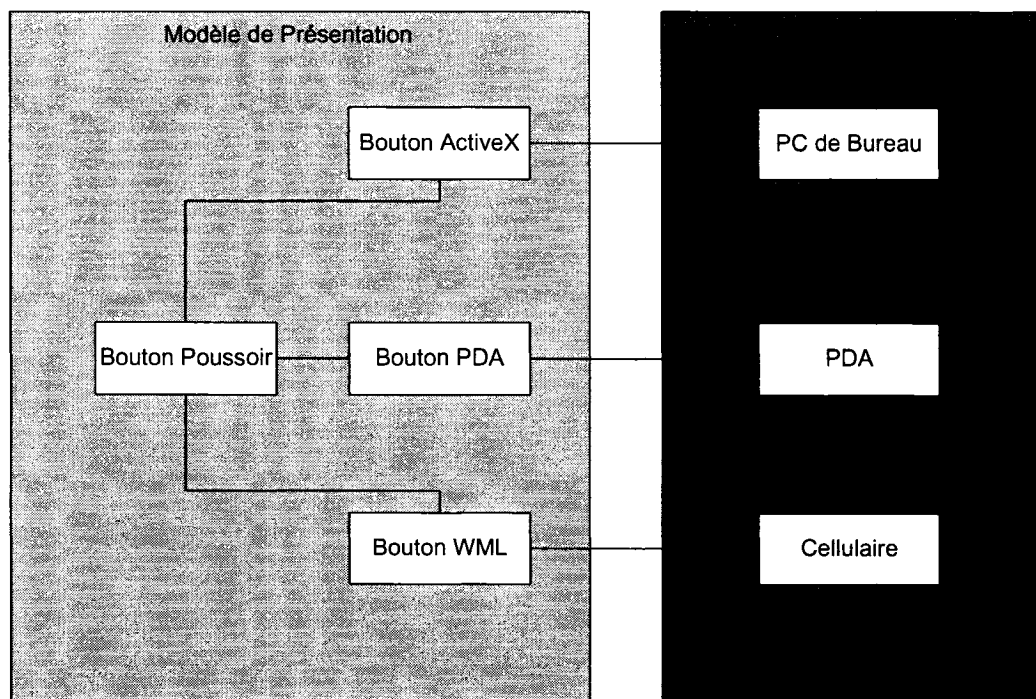


Figure 2.7 Adaptation des objets d'interface

Les AIO ne constituent qu'un élément du puzzle permettant de concevoir des applications adaptables. En effet, les AIO ne prennent pas en considération toutes les contraintes liées au développement des applications avec des interfaces usager adaptables et s'exécutant sur des appareils mobiles. Ce concept d'implémentation d'application adaptable peut être implémenté à l'aide d'outils tels XML.

2.7.2 Description de XML

XML (O'Reilly, 2003) offre la possibilité d'être aussi bien compréhensible par les acteurs humains que par les ordinateurs. En effet, sa structure permet aux humains, en l'occurrence aux concepteurs d'applications, de caractériser une interface usager haut niveau. XML permet aux machines de traiter cette description sans pour autant qu'il y ait besoin d'une intervention humaine. Il permet aussi de créer des formats d'information et de partager cette information ainsi que son format sur un réseau (Web,

intranet, etc.). La notation utilisée est sous forme d'étiquettes qui sont faciles à lire et à comprendre par les humains.

Les différents objets d'interface qui peuvent être utilisés incluent les objets pour les choix multiples, les lignes, les entrées de texte, les poussoirs et les sorties telles l'audio et la vidéo. Ceux ci peuvent être composés pour représenter un nouvel objet d'interface avec des fonctionnalités combinées. Les deux étiquettes les plus couramment utilisées sont "group" et "action". L'étiquette "group" permet de grouper des objets qui n'ont pas de signification quand ils sont séparés. Et l'étiquette "action" permet au concepteur de spécifier quelle action effectuer si l'objet d'interface, qui est le nœud parent dans l'arborescence XML, est sollicité (Luyten et al. 2002). Nous pouvons prendre comme exemple, trois objets d'une interface usager sensée contenir le jour, le mois et l'année d'une date, comme à la Figure 2.8. Il faudra que l'application se trouvant à la destination ait des informations quant à la relation qui existe entre ces trois objets afin de garder la logique dans la représentation de ces éléments.

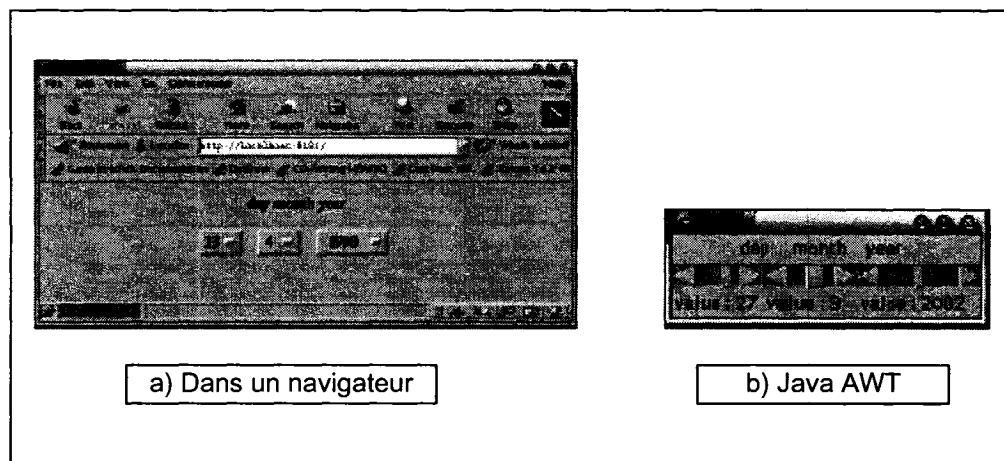


Figure 2.8 Deux vues différentes générées automatiquement

2.7.3 Exemples d'implémentation

La manière dont une interface usager peut être diffusée dans un langage de description d'interfaces usager en temps réel a été décrite par Luyten et al. (2002a), et

par Luyten et al. (2002b). En effet, le langage XML est utilisé afin d'avoir un certain niveau d'abstraction de l'interface à diffuser, et celui-ci permet aussi de transporter cette information à travers un réseau. Ainsi XML offre une abstraction de l'interface usager en utilisant les AIO. Lors de l'envoi, l'interface usager est transformée en un fichier XML à l'aide de XSLT (eXtensible Stylesheet Language Transformations) qui est un standard pour décrire la transformation de la structure d'un fichier XML et qui change ainsi les objets de l'interface qui peuvent être considérés comme des CIO en AIO. Puis ce fichier est transféré à travers le réseau. Ainsi une fois à destination, l'éditeur de l'interface usager peut choisir de présenter d'une manière différente l'interface tout en gardant les mêmes fonctionnalités. Il devra avoir des informations quant au contexte de présentation de ces fonctionnalités.

Une fois que l'UI a été converti en XML, lors de l'envoi et de la réception, une transformation est faite en utilisant XSLT qui se charge d'adapter l'interface usager aux contraintes de la nouvelle plate-forme. Donc pour chaque plate-forme, un fichier XSLT est défini et se charge de l'adaptation de l'interface.

Une autre solution a été proposée par Mitrović et Mena (2002) et est basée sur les agents mobiles et sur XUL (eXtensible User interface Language). XUL est un langage conçu pour la définition d'interfaces usagers portables. jXUL (java XUL) (Sourceforge, 2003) permet d'interpréter les fichiers XUL pour donner exclusivement une interface Java Swing. Afin de permettre de générer d'autres interfaces, XSLT a été utilisé pour générer des interfaces HTML (HyperText Markup Language) et WML à partir des fichiers XUL. Un agent mobile est une entité logicielle à qui est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome et en coopération avec d'autres agents. Un agent est mobile, social, animé par un objectif, réactif et s'adapte à son environnement. Ces différentes technologies ont été mises en commun afin de construire un agent mobile qui soit capable de s'adapter à la plate-forme sur laquelle il se trouve et de pouvoir générer une interface qui peut être sous la forme Java Swing, HTML ou WML.

D'après les résultats obtenus lors de l'étude, il a été montré que le système est stable, capable de s'adapter à différentes plates-formes et, parce qu'il est basé sur des agents mobiles, s'avère très intéressant pour les applications dans des environnements mobiles.

Nous avons vu dans ce chapitre que l'implémentation d'un mécanisme d'accès à l'information dépendant du contexte peut être liée à la plate-forme logicielle, aux périphériques de communication des infrastructures mobiles, aux protocoles mobiles et aux infrastructures des réseaux mobiles. Les différentes méthodes présentées n'ont pas été conçues pour la conception spécifique d'une plate-forme d'apprentissage mobile. Nous allons nous pencher dans le chapitre suivant sur l'exploitation des moyens de communication et de calcul des appareils des étudiants mobiles afin de concevoir une plate-forme d'apprentissage qui puisse être capable de détecter son contexte d'utilisation. Cette plate-forme permettra la collaboration entre plusieurs étudiants ou entre les étudiants et le professeur, par l'échange d'informations de cours.

CHAPITRE III

MODÈLE ET ARCHITECTURE D'ACCÈS À L'ENVIRONNEMENT D'APPRENTISSAGE

Depuis l'introduction du WAP (Wireless Application Protocol), beaucoup d'intérêt a été porté sur les possibilités de l'Internet mobile. De plus, l'apprentissage mobile est perçu comme la prochaine étape après l'apprentissage électronique. La plupart des recherches qui ont été effectuées dans le domaine de l'apprentissage électronique ont abouti à des applications conçues pour être exécutées sur des ordinateurs de bureau. Cependant, les étudiants sont de plus en plus mobiles et l'utilisation des appareils sans fil est devenue très courante pour les étudiants qui sont partagés entre leurs cours à l'université, et leur emploi à temps plein ou à temps partiel, ou leurs autres activités. Ce nomadisme des étudiants et leur goût pour les ordinateurs de poche a poussé certains chercheurs à développer des applications destinées à ces plateformes mobiles qui sont caractérisés par leur propres contraintes. Ces appareils offrent tout de même un grand potentiel de connectivité avec l'avènement des réseaux sans fil de troisième génération et des protocoles de communication tels que le Bluetooth et l'infrarouge. Ces moyens de communications permettent à ces appareils mobiles, tels les PDA, ordinateurs de poche et téléphones cellulaires, de se connecter partout et en tout temps. Dans ce chapitre, nous définissons et situons clairement le problème des mécanismes d'accès à l'information dépendants du contexte pour des étudiants nomades, puis nous définissons les requis liés à ce problème. Par la suite, nous présentons un modèle d'apprentissage mobile adaptable. Avant de conclure avec les cas d'utilisation de notre architecture, nous traiterons des technologies utilisées pour le développement de notre application.

3.1 Adaptabilité des applications mobiles d'apprentissage

L'adaptabilité des applications d'apprentissage mobiles est définie comme la possibilité, pour ces applications, de prendre en compte leur contexte d'exécution et d'utilisation afin de rendre disponibles des données pédagogiques (Wang, 2004). Nous désignons par contexte d'exécution, les ressources matérielles et logicielles mises à la disponibilité de l'application telles que la taille de l'écran de l'appareil, sa puissance de calcul, ses périphériques d'entrée/sortie et ses protocoles de communication. Le contexte d'utilisation quant à lui définit tous les facteurs externes à l'environnement d'exécution de l'application et qui auraient une incidence sur ses fonctions, à savoir les caractéristiques de l'utilisateur et le lieu d'exécution de l'application. L'environnement d'exécution des applications mobiles est très limité. En effet, les appareils mobiles disposent généralement d'une source d'énergie limitée puisque la plupart fonctionnent avec une batterie garantissant peu d'autonomie. De plus, ces appareils offrent peu de puissance de calcul, et ont une capacité d'affichage réduite. Les applications d'apprentissage mobile doivent donc prendre en compte l'ensemble de ces contraintes afin de pouvoir s'exécuter et être utilisées par les étudiants.

Dans cette section, nous allons définir le problème de l'adaptabilité des applications d'apprentissage mobile, puis identifier les requis d'une telle adaptabilité.

3.1.1 Définition du problème

Supposons un cours avec un professeur et ses étudiants. L'approche proposée par l'apprentissage électronique est de disposer d'une page web contenant les informations relatives au cours, à savoir les notes de cours, les travaux à effectuer, les dates des examens et de remise des travaux, le tout, mis à jour par le professeur. L'étudiant doit donc souvent aller sur cette page web pour se tenir informé des dernières modifications relatives au cours. Un moyen utilisé pour tenir les étudiants au courant des mises à jour est le courriel, mais cette manière de procéder suppose encore que les étudiants aillent régulièrement vérifier le contenu de leur boîte de courriel. Les étudiants peuvent donc ne

pas recevoir les messages de leur professeur ou les dernières informations concernant leurs travaux.

Supposons maintenant qu'un étudiant veuille, à la veille d'un examen, rencontrer son professeur pour lui poser une question. Même si le professeur a indiqué ses disponibilités sur la page web du cours, il peut s'absenter momentanément de son bureau. Il est courant de voir, à la veille d'examens, plusieurs étudiants devant le bureau d'un professeur l'attendant pour lui poser des questions alors qu'en collaborant, ces étudiants s'apercevraient qu'ils pourraient obtenir la réponse à leur question d'un autre étudiant attendant le professeur.

Comme nous l'avons vu dans le chapitre précédent, plusieurs travaux essaient d'apporter une solution au problème de l'adaptabilité des applications mobiles en fonction de leur contexte d'utilisation. Cependant, ces recherches ne se concentrent pas sur les mécanismes d'accès à l'information en fonction du contexte pour les applications d'apprentissage. Ils ne prennent en considération ni le contexte d'exécution et d'utilisation, ni le besoin de communication et de transfert de données entre étudiants/étudiants ou entre étudiants/professeurs. Chen et Kotz (2000) ont fait un rapide tour des travaux qui sont actuellement en cours dans le domaine de l'adaptabilité des applications en fonction du contexte. Cependant, nous constatons que les mécanismes de détection de contexte d'utilisation utilisés sont basés sur des technologies telles que le GPS qui est rarement disponible pour les appareils tels que les téléphones cellulaires ou utilisent des solutions propriétaires qui ne sont pas implémentées par la plupart des fabricants d'appareils mobiles. De plus, les modèles présentés ne prennent pas aussi bien en considération le contexte d'exécution, que celui d'utilisation des applications. Nous proposons dans ce mémoire de développer une architecture adaptable à la plate-forme d'exécution et au lieu d'utilisation, pour l'apprentissage mobile, et utilisant des technologies largement implémentées dans le marché des appareils mobiles. Notre modèle est utilisable aussi bien à l'extérieur qu'à l'intérieur d'un bâtiment.

3.1.2 Analyse des requis du contexte d'exécution et d'utilisation

Nous pouvons identifier une série de requis qui caractérisent l'ensemble des moyens d'apprentissage. Ces requis ont servi de base pour le développement et l'analyse de modèles supportant l'adaptabilité dans le domaine de l'apprentissage mobile :

- le contenu éducatif est dynamique : les étudiants ont besoin de données mises à jour et ont besoin de communiquer avec des sources d'information variées ;
- le contenu éducatif est opéré en temps réel : les étudiants peuvent avoir besoin de l'information de n'importe où et en tout temps ;
- le contenu éducatif est collaboratif : étant donné que les étudiants peuvent apprendre les uns des autres, les applications d'apprentissage doivent pouvoir les connecter à des bases de données, à des professeurs, des experts ou d'autres étudiants ;
- le contenu éducatif est personnel : chaque étudiant doit pouvoir sélectionner l'information dont il a besoin à un certain moment ;
- le contenu éducatif est complet : les étudiants doivent avoir la possibilité de recueillir l'information dont ils ont besoin à partir de plusieurs sources.

Étant donné que notre but est la conception d'un modèle d'apprentissage mobile qui prend en considération aussi bien le contexte d'utilisation, que le contexte d'exécution, les requis de notre application dépendent de la plate-forme d'exécution, de son environnement d'exécution et de son utilisateur qui peut être un étudiant ou un professeur. Pour s'exécuter sur plusieurs plate-formes, notre modèle d'apprentissage doit prendre en considération les caractéristiques techniques des appareils mobiles (Gaedke, et al., 1998) :

- La taille des écrans : Il faut tester les applications d'apprentissage sur plusieurs écrans d'appareil mobile afin de vérifier que leur interface usager s'affiche convenablement. Les différentes pages de l'application doivent contenir de l'information succincte. Le défilement d'écran peut être fastidieux sur un

appareil sans fil. De ce fait, les pages doivent être les plus courtes possibles. Il faut éviter les éléments larges et les tables, dont la taille dépasse celle de l'écran, parce que la plupart des appareils mobiles ne permettent pas un défilement horizontal d'écran. La taille des tableaux ou des images ne doit pas excéder la taille de l'écran ;

- La consommation d'énergie. La plupart des appareils mobiles fonctionnent avec des batteries. L'implémentation du modèle d'apprentissage mobile ne devrait pas être énergivore. L'utilisation prolongée de connexions fait partie de ce qui consomme le plus d'énergie pour les appareils mobiles. Notre application d'apprentissage mobile ne devrait donc pas avoir de temps de connexion trop longs ;
- La puissance de calcul. Les applications doivent effectuer le minimum de calculs sur la plate-forme mobile. En cas de calculs complexes, un serveur s'en occupera et fournira à l'application les pages à afficher ;
- Les capacités de communication. Certains types de communication peuvent coûter cher. Il faut donc utiliser les technologies les moins coûteuses afin d'accommoder les utilisateurs. Parmi les protocoles les mieux appropriés à cette fin, on retrouve les connexions infrarouges et Bluetooth ;
- La Navigation. Notre modèle doit permettre une navigation efficace. Cela inclut la possibilité, pour l'utilisateur, de revenir au menu principal de son appareil mobile s'il veut exécuter d'autres fonctions telles que la sauvegarde des données.

Notre modèle d'apprentissage est aussi valide pour un étudiant que pour un professeur. Notre architecture d'accès à l'information dépendant du contexte devrait aussi prendre en considération son environnement d'exécution ainsi que son utilisateur qui introduit des contraintes de conception :

- Sélection de l'information. Les appareils mobiles n'offrent pas le même confort de manipulation que les ordinateurs de bureau. Seules les informations essentielles doivent être envoyées aux étudiants à travers leurs appareils mobiles.

L'information la plus importante doit être mise en relief afin que l'utilisateur puisse rapidement prendre connaissance de l'objet du message reçu ;

- Mobilité de l'utilisateur. Il faut considérer les possibles changements de contexte. L'étudiant ou le professeur peut être dans une salle de classe, dans un bus ou à la maison. Les besoins d'information des utilisateurs changent en fonction de leur localisation géographique. Un étudiant rentrant à la maison, à la veille d'un cours, a besoin de connaître la salle de classe où se tiendra celui-ci, et le sujet sur lequel il portera ainsi que les différentes notes à réviser pour le lendemain.

Ces requis nous serviront de ligne directrice pour la conception d'un modèle d'application adaptable supportant la mobilité des étudiants et des professeurs.

3.2 Architecture d'un modèle d'apprentissage mobile adaptable

Après avoir fait la revue des requis et des contraintes, nous proposons dans cette section un modèle d'apprentissage mobile supportant l'adaptation à son contexte. Nous commençons par présenter les caractéristiques des appareils mobiles pour lesquels l'architecture a été conçue, ensuite nous présentons les différentes technologies utilisées pour l'implémentation de notre modèle, et enfin nous décrivons l'architecture développée.

3.2.1 Conceptualisation de l'environnement d'apprentissage mobile

Comme nous l'avons vu, les plates-formes mobiles ont des caractéristiques que nous devons prendre en considération.

Caractéristiques des appareils mobiles

L'apprentissage mobile serait impossible sans l'utilisation d'appareils mobiles. Leurs caractéristiques varient beaucoup de par leurs fonctionnalités, leur taille et leur prix. Ces appareils sont de plus en plus communs dans l'environnement technologique

actuel. Beaucoup d'étudiants disposent au moins d'un ordinateur portable, d'un agenda de type Palm ou d'un téléphone cellulaire. Il existe actuellement plus d'un demi-milliard de téléphones cellulaires disposant de la technologie Internet de par le monde. À partir de l'année 2005, le nombre d'appareils mobiles a dépassé le nombre d'ordinateurs de bureau vendus. Plusieurs types d'appareils sans fil sont utilisés dans l'apprentissage mobile (Georgiev et al., 2004) :

- Les PDA (Personal Digital Assistant) sont des ordinateurs miniatures supportant la mobilité. Ils sont extensibles de par la possibilité de leur connecter un clavier, leur ajouter de la mémoire supplémentaire, une antenne ou d'installer des logiciels.
- Les téléphones cellulaires, quant à eux, utilisent l'Internet mobile pour échanger des messages vocaux, des courriels, et des pages web miniatures partout et à tout moment. Les téléphones cellulaires compensent leur lacunes par rapport aux PDA en utilisant l'Internet ou d'autres moyens de connexion. Les étudiants peuvent apprendre une langue étrangère en utilisant un PDA et un programme dans une mémoire extensible, tandis qu'en utilisant un téléphone cellulaire, ils pourraient avoir accès à la même information située sur une base de données Internet. De plus, les téléphones cellulaires ont l'avantage de coûter moins cher que les PDA. Ils sont donc plus accessibles pour le budget des étudiants.
- Les ordinateurs portables ont les mêmes capacités qu'un ordinateur de bureau. D'une manière générale, ils peuvent être branchés physiquement à un réseau ou être utilisés sans branchement physique au réseau électrique et avec une connexion sans fil. Ils sont beaucoup plus petits que les ordinateurs de bureau avec un écran plus petit. Les applications qui s'exécutent sur les ordinateurs de bureau peuvent aussi bien s'exécuter sur les ordinateurs portables sans pour autant devoir s'adapter et vice versa. Le prix des ordinateurs portables est toujours plus élevé que les ordinateurs de bureau disposant de capacité équivalente.

- Les téléphones intelligents sont des appareils hybrides qui combinent les fonctionnalités des téléphones cellulaires et des PDA. Ils ont une taille plus réduite que les PDA mais sont un peu plus grands que les téléphones cellulaires. D'habitude, ils ne disposent pas d'un clavier complet mais peuvent être en mesure de reconnaître l'écriture de leur utilisateur. Ils utilisent des systèmes d'exploitation tels que Symbian[®] (SYMBIAN), Windows Mobile[™] (WINMOB) ou d'autres systèmes d'exploitation équivalents. Étant donné qu'ils sont dotés de fureteur, ils peuvent fournir du contenu multimédia.

Les appareils mobiles disposent de plusieurs des fonctionnalités des ordinateurs de bureau. Cependant, plusieurs limitations les empêchent de les remplacer, à savoir la bande passante qui leur est accessible, le coût des services de communication, la vitesse d'entrée de données et la vitesse de leur processeur. La vitesse de transmission et la couverture sont les deux facteurs les plus importants des réseaux de données destinées aux appareils sans fil. Étant donné que la plupart des applications d'apprentissage mobile requièrent l'échange d'informations, les SMS (Short Message Service) des réseaux GSM (Global System for Mobile communication) constituent une alternative. Cependant, l'utilisation des SMS ou des MMS (Multimedia Message Service) est soumise à des frais d'envoi par la plupart des fournisseurs d'accès des réseaux de téléphonie cellulaire. L'utilisation de ce type de technologie pour les étudiants pourrait être coûteuse. Un autre facteur important est le support du multicast où la communication entre plusieurs membres d'un groupe de travail est nécessaire pour des applications nécessitant la collaboration de plusieurs personnes telles que les réseaux de discussions. Nous devons donc à travers notre application supporter le multicast afin de permettre une meilleure communication entre étudiants. L'apprentissage mobile requiert une qualité de service qui affecte la performance des applications d'apprentissage en termes de délai, de perte, et autres attributs de la qualité de service.

Les appareils mobiles utilisent déjà plusieurs systèmes d'exploitation, à savoir Palm OS, Microsoft Pocket PC (Windows CE), et Symbian EPOC. Tous ces systèmes

ont l'avantage d'avoir une plate-forme d'exécution Java. Celle ci nous permet de faire abstraction du type de téléphone utilisé et de nous concentrer sur notre modèle d'application. Nous proposons d'utiliser les avantages de portabilité de Java pour permettre à notre application de s'exécuter sur plusieurs plates-formes ayant des caractéristiques différentes, tout en tirant avantage des propriétés de ces plate-formes.

Protocoles de communication mobiles

Quelques nouveaux appareils destinés aux réseaux de troisième génération disposent de la technologie nécessaire au visionnement de vidéo mais le coût de telles communications constitue encore un frein à son utilisation pour ou par les étudiants dans un contexte d'apprentissage mobile. Il existe plusieurs moyens de connexion aux réseaux sans fil (Georgiev et al., 2004).

- Le GSM est un réseau qui utilise la large bande TDMA (Time Division Multiple Access). Lancé en Europe comme un standard de téléphonie mobile numérique, le GSM a une présence dans plus de 100 pays. Le réseau GSM opère dans la bande de 900 MHz et du 1800 MHz en Europe, en Asie et en Australie, et dans la bande de 1900 MHz en Amérique du Nord et dans certains pays d'Amérique du Sud et d'Afrique. Il permet l'intégration de la messagerie audio, du transfert de données, du service de SMS, ainsi que des communications sécurisées. Il offre la meilleure qualité de la voix transmise que tous les autres réseaux de communication numérique sans fil.
- Le Wireless Application Protocol (WAP). C'est un protocole de communication sans fil gratuit. Il permet la création de services de communications avancées et l'accès à des pages Internet à travers un téléphone cellulaire. Le WAP est de facto le standard de l'industrie supporté par la plupart des fournisseurs. Les appareils WAP supportent le langage WML (une application XML) qui a été optimisé pour les écrans de petite taille et la navigation à travers de pages sans l'aide d'un clavier. Le WAP supporte aussi le langage de script WMLScript.

- Le General Packet Radio Service (GPRS). Le GPRS est un protocole de communication à commutation de circuit permettant la connexion haute vitesse sans fil à Internet. Le GPRS offre une connexion quatre fois plus rapide que la connexion offerte par le GSM. En ce moment, plus de 300 opérateurs de téléphonie cellulaire offrent un service commercial GPRS à travers le monde.
- Le Bluetooth est une technologie radio de connexion sans fil à courte portée. Bluetooth permet la connexion sans fil et le transfert de données entre des téléphones cellulaires, des ordinateurs de poche et des PDA. Ceci permet la simplification des communications et la synchronisation entre les différents appareils. Le protocole Bluetooth permet la formation de réseaux ad-hoc dynamiquement avec des appareils détenant cette technologie. Le protocole a été développé en prenant en considération la robustesse et le coût. Cette technologie cible les utilisateurs mobiles qui ont besoin d'établir des connexions rapides ou de petits réseaux entre leur ordinateur, leur appareil cellulaire et d'autres périphériques. L'actuelle portée des antennes Bluetooth est de 10 m. Bluetooth opère dans la bande de fréquence de 2.4 GHz avec un débit de transmission de 1 Mbit/seconde. Un seul appareil mobile doté de la technologie Bluetooth peut opérer avec un taux de transfert de données de 721 kbits/seconde ou un maximum de 3 canaux de voix. Bluetooth offre un niveau de sécurité aussi bien au niveau de la couche application que de la couche physique. Cela veut dire que pour chaque unité Bluetooth, l'authentification et l'encryption des données sont implémentées de la même manière (Haartsen et al., 1998).
- Le IEEE 802.11 est un type de technologie radio utilisé pour les réseaux locaux sans fil. C'est un standard qui a été développé par l'IEEE (Institute of Electrical and Electronic Engineers). Le Wi-Fi est composé de plusieurs standards opérant dans différentes fréquences radio. Le 802.11b est un standard pour les réseaux locaux et opère dans la bande de fréquence de 2.4 GHz avec une largeur de bande de 11 Mbps; le 802.11a est un standard différent pour les réseaux locaux sans fil et concerne les systèmes opérant dans la bande de fréquence de 5 GHz

avec une largeur de bande de 54 Mbps. Le 802.11g est un autre standard pour les réseaux locaux sans fil opérant dans la bande de 2.4 GHz mais avec une largeur de bande de 54 Mbps. Peu d'appareils mobiles disposent de ce moyen de communication.

- Le Infrared Data Association (IrDA) est une association qui définit une suite de protocoles pour l'échange de données infrarouge entre deux appareils distants de 20 à 30 centimètres. Les appareils disposant de la technologie IrDA peuvent envoyer des données à un débit de 115.2 Kbps à 4 Mbps. La plupart des téléphones intelligents, PDA, imprimantes, et ordinateurs portables, utilisent la technologie IrDA.

La plupart de ces réseaux de communication remplissent les requis d'implémentation de notre modèle d'apprentissage mais induisent des coûts qui pourraient constituer un frein à leur utilisation. Nous avons choisi d'utiliser la technologie Bluetooth qui répond à tous nos requis de connexion et qui a l'avantage de permettre des communications sans frais. En utilisant cette technologie, les utilisateurs d'appareils mobiles peuvent facilement synchroniser leur information avec des ordinateurs de bureau ou des ordinateurs portables. Ce support d'information permet aussi d'envoyer des messages textes, des fax, d'initier des impressions et d'avoir toutes les applications d'apprentissage mobile totalement coordonnées. Nous allons utiliser la portée de ce protocole, qui peut être perçue comme une faiblesse. Mais dans notre cas, nous allons la considérer comme un avantage pour la localisation des appareils mobiles et des terminaux. En effet, Bluetooth nous permet de localiser les étudiants nomades avec une précision de 10 mètres. Nous allons l'utiliser dans le cadre de notre application d'apprentissage mobile comme un moyen de localiser les étudiants ou les professeurs, mais aussi comme un moyen de transférer des données.

Le contexte d'utilisation

La conscience du contexte d'utilisation des appareils mobiles est un paradigme dans lequel les applications peuvent découvrir et prendre avantage de l'information contextuelle (comme le type d'appareil utilisé ainsi que ses ressources, l'endroit d'utilisation, les personnes et les appareils à proximité, et l'activité de l'utilisateur).

Même si l'information du contexte d'utilisation est disponible, son bon usage reste un défi pour les concepteurs d'application. Le type d'adaptation des applications conscientes de leur contexte peut être divisé en plusieurs catégories (Chen et Kotz, 2000) :

- le choix du plus proche : une technique d'interface usager où les objets à proximité sont mis en exergue;
- la reconfiguration contextuelle automatique : un processus pour ajouter de nouvelles composantes, enlever des composantes existantes ou changer les connexions entre composants en fonction des changements de contexte;
- les informations et les commandes contextuelles : elles peuvent produire différents résultats selon le contexte dont elles sont issues;
- les actions déclenchées par le contexte : de simples règles SI-ALORS qui définissent les choix à effectuer afin de s'adapter au contexte.

Pour notre modèle, nous allons combiner différents types d'adaptabilité au contexte, à savoir le choix du plus proche où les appareils mobiles à proximité vont pouvoir se connecter. De plus, nous allons utiliser les informations contextuelles qui vont permettre à notre application d'exécuter des actions SI-ALORS. Ces dernières différencieront les cas d'utilisation de l'application, que cela soit par un professeur ou par un étudiant.

Dans notre cas, nous voudrions connaître l'emplacement d'un étudiant, par exemple, pour lui transmettre des données à son arrivée ou à son départ de l'université. Étant donné que le lieu d'utilisation est un contexte important et changeant avec la mobilité de l'utilisateur, un système de localisation constitue un élément critique de

notre application dont le fonctionnement dépend de son contexte géographique d'utilisation. Il est simple d'obtenir la position d'un appareil mobile avec la collaboration de son utilisateur. Cependant, l'utilisateur est la plupart du temps distrait lors de ses déplacements. Nous allons donc utiliser le protocole Bluetooth pour déterminer l'emplacement de l'unité mobile sans pour autant que l'utilisateur n'ait à le fournir de manière explicite à notre application d'apprentissage. Ceci constitue un avantage, étant donné que l'utilisateur n'aura pas à spécifier son emplacement à l'application quand il change de lieu géographique.

3.2.2 Plate-forme de développement des applications d'apprentissage mobiles

Notre modèle d'application d'apprentissage mobile adaptable s'exécute sur une plate-forme matérielle, mais aussi logicielle. Nous allons dans cette section faire un survol de la plate-forme Java adaptée aux appareils portables, et ainsi justifier son utilisation pour notre application.

La plate-forme JAVA

Les appareils mobiles contemporains intègrent un appareil photo, un lecteur mp3 ou la radio. L'évolution de la téléphonie est loin de s'essouffler. Cependant, les téléphones évoluent mais sur des standards différents selon les fabricants et les modèles. Le développement d'application passe en général par l'utilisation d'une API propriétaire souvent écrite en C ou C++. Le portage d'une application développée en C++ implique donc l'adaptation du code pour presque chaque modèle de téléphone. Les coûts de production augmentent donc forcément. La plate-forme Java constitue une solution au problème de portabilité des applications mobiles.

Java est d'abord connue comme un environnement de programmation de serveur centré sur les technologies qui ont généré le Java 2 Entreprise Edition (J2EE), telles que les JavaBeans (EJB), les servlets (SERVLET) et les JavaServer Pages (JSP). Ce qui est appelé familièrement Java est plus formellement connu sous le nom de Plate-forme Java 2. Elle est divisée en trois éditions : Edition Standard Java 2 (J2SE), Edition Entreprise

Java 2 (J2EE) et Edition Micro Java 2 (J2ME). Chaque édition de la plate-forme procure toutes les classes d'interface usager de Java. J2ME est utilisé pour des applications devant s'exécuter sur des plates-formes mobiles.

Dans J2ME, l'environnement d'exécution de Java est adapté aux capacités des appareils mobiles de petite dimension avec peu de ressources comparativement aux ordinateurs de bureau ou aux serveurs. Dans le cœur de l'édition Micro Java 2, il y a trois concepts de base : les configurations, les profils et les bibliothèques optionnelles. Ces concepts déterminent les fonctionnalités de Java et les interfaces de programmation d'application qui sont disponibles, ainsi que la façon dont les applications sont encapsulées en bibliothèque.

Les appareils mobiles supportant J2ME sont de nature très différente, J2ME définit alors deux types de spécifications fonctionnant conjointement, les configurations et les profils. Une configuration est un environnement d'exécution complet composé :

- d'une machine virtuelle Java pour exécuter le code Java ;
- du code natif pour interfacier avec le système d'exploitation utilisé ;
- d'une bibliothèque de classes Java.

Les configurations ne définissent aucune classe d'interface usager. J2ME définit deux configurations, la CLDC et la CDC. La première est utilisée pour les appareils de performances très limitées, c'est à dire avec peu de mémoire et de puissance de calcul. La CDC inclut, quant à elle, une machine virtuelle Java complète et un nombre plus important de classes du noyau. De ce fait, elle requiert plus de mémoire et un processeur plus rapide que pour la CLDC.

Lorsqu'une configuration définit le fondement d'une application, un profil en fournit la structure. Les profils définissent l'ensemble des API à utiliser dans une application J2ME et sont conçus spécialement pour chaque configuration. Un profil des classes doit être implémenté afin de remplir les fonctionnalités manquantes et de supporter les fonctionnalités spécifiques des appareils mobiles. Par exemple, la plupart des profils définissent les classes des interfaces usager pour exécuter les applications

interactives. Il existe plusieurs profils dans diverses étapes du développement. Le premier profil est le Mobile Information Device Profile (MIDP), un profil basé sur la CLDC pour exécuter des applications sur des téléphones cellulaires et les pagettes interactives avec de petits écrans, la connectivité http sans fil, et les plates-formes avec des mémoires limitées. Un autre profil basé sur la CLDC et en cours de développement est le Personal Digital Assistant Profile (PDAP). Il étend le MIDP à d'autres classes et fonctionnalités additionnelles pour des appareils sans fil plus puissants. En termes de profils basés sur la CDC, le Foundation Profile (FP) étend le CDC avec des classes additionnelles de J2SE. Les machines virtuelles pouvant être utilisées dans un environnement J2ME sont la KVM (Kilobyte Virtual Machine) pour la CLDC et CVM (C-Virtual Machine) pour la CDC. Ces machines virtuelles ont été spécifiquement développées pour supporter l'exécution d'applications dans des environnements contraignants, en termes de performances, tels que les appareils mobiles. Elles facilitent également la portabilité des applications sur différentes plate-formes.

Java comporte aussi des bibliothèques optionnelles. Ce sont une série d'API, en support à des fonctionnalités additionnelles et communes qui n'appartiennent à aucune configuration ou profil spécifique. Le support Bluetooth par exemple est défini comme une bibliothèque optionnelle. Les bibliothèques optionnelles ont leurs propres requis de base comme les configurations et les profils. Les bibliothèques optionnelles ont aussi des dépendances spécifiques sous une configuration. Elles ne définissent pas un environnement d'exécution complet mais juste un ensemble d'API.

3.3 Modèle d'application d'apprentissage mobile adaptable

Notre modèle d'apprentissage mobile adaptable a pour but de montrer la possibilité d'adaptation des applications d'apprentissage mobile en fonction de leur contexte d'utilisation. Dans ce cas, nous appelons contexte le lieu d'utilisation de l'application.

Comme nous l'avons vu dans les paragraphes précédents, la plate-forme J2ME développée par SUN et destinée aux plates-formes mobiles, ainsi que la technologie

Bluetooth ont l'avantage d'être des technologies éprouvées et supportées par un large éventail de fabricants d'appareils mobiles. Nous allons utiliser la configuration CLDC et le profil MIDP. Notre but est de développer une application utilisable par un maximum d'appareils mobiles possible. Les étudiants disposant d'un budget limité, nous allons prendre pour acquis que le type de téléphone utilisé possède beaucoup de contraintes mais supporte au minimum la plate-forme J2ME et Bluetooth comme montré à la Figure 3.1. Nous allons implémenter notre modèle en utilisant l'environnement de développement intégré NetBeans (NETBEANS) conçue par SUN. Nous allons y rajouter le "NetBeans Mobility Pack" qui permet de concevoir des applications spécifiquement destinées aux appareils mobiles.

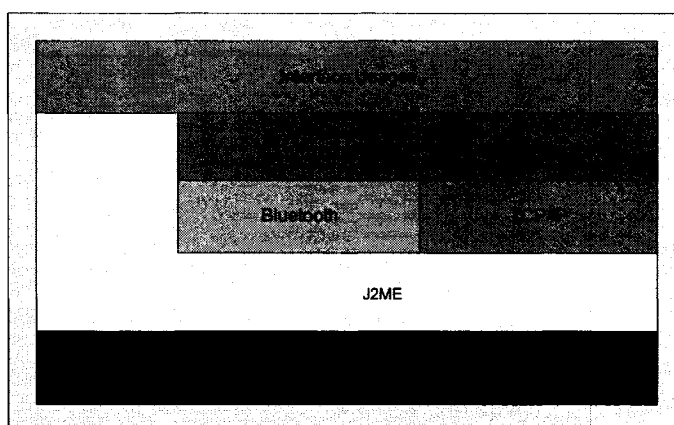


Figure 3.1 Couches de l'application

Notre allons donc implémenter un modèle incluant les technologies J2ME et Bluetooth afin de répondre aux requis d'adaptabilité de notre application, et de pouvoir détecter le contexte d'exécution ainsi que le contexte géographique d'utilisation de notre application. Nous pourrions détecter l'utilisateur de notre application par son identification qu'il nous fournira lors de l'exécution de notre application d'apprentissage. Notre modèle prendra aussi en compte le profil de l'utilisateur. Les scénari d'utilisation seront aussi implémentés en fonction du type d'utilisateur.

3.3.1 Cas d'utilisation des applications d'apprentissage mobile adaptables

Notre objectif est de montrer qu'une application d'apprentissage peut reconnaître son contexte géographique d'utilisation et effectuer des tâches en conséquence. Pour ce faire, nous avons développé une application basée sur notre modèle et implémentée dans un appareil mobile doté de la technologie Bluetooth. Ce terminal peut être vu comme un ordinateur muni d'une ou de plusieurs antennes Bluetooth ou d'un autre appareil mobile disposant d'une antenne Bluetooth. Notre application sera en mesure de se connecter automatiquement à la maison, à l'université ou avec d'autres appareils mobiles pour échanger de l'information, et recevoir des fichiers de l'Internet en fonction du lieu de connexion et de son utilisateur. Le fichier peut être un fichier texte, un fichier image ou une page XML. Notre application peut aussi bien être utilisée par des étudiants que par des professeurs. Nous montrons à la Figure 3.2 les différents cas d'utilisation de notre application d'apprentissage mobile adaptable.

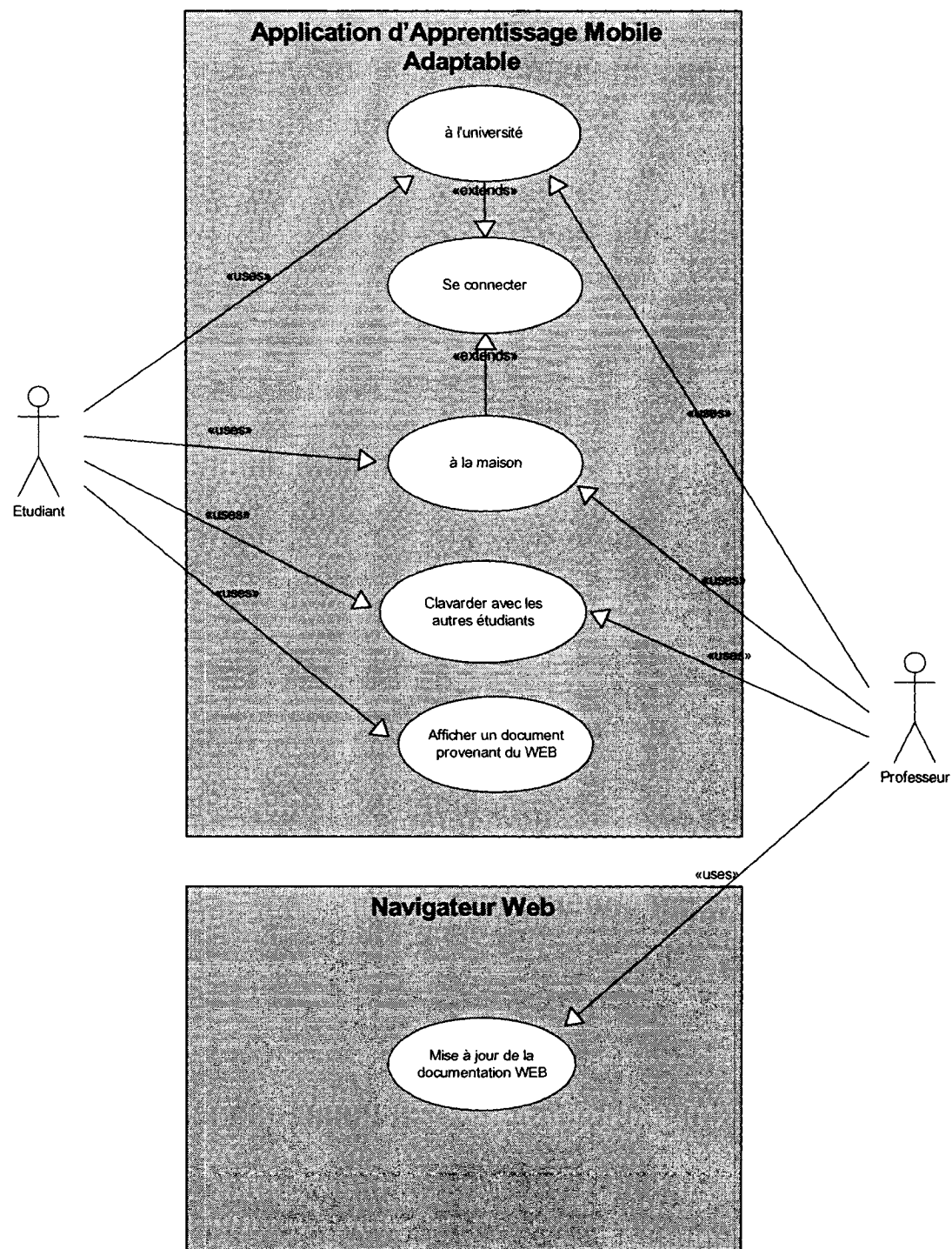


Figure 3.2 Cas d'utilisation de notre application d'apprentissage mobile

Notre application aura un comportement différent selon son contexte géographique et son utilisateur. Nous avons identifié deux types d'utilisateur et trois contextes d'utilisation pour chacun d'eux. Ces types d'utilisateurs sont les étudiants et les professeurs. Les premiers sont définis ici comme des membres d'un cours et qui sont supposés recevoir de l'information sur leur cours. Leur support de communication utilisé est le site web de leur cours qu'ils vont visiter de manière épisodique. Notre but est donc de donner de l'information contextuelle aux étudiants au moment opportun à travers de courts messages qui leurs sont envoyés. Ces étudiants pourront se servir de leur appareil mobile pour se connecter automatiquement au serveur de leur cours afin de recevoir de l'information en fonction de leur lieu géographique. Notre modèle pourra aussi être utilisé par les professeurs afin de leur permettre une diffusion de message aux membres de leur cours. Elle leur permettra ainsi d'être plus proches de leurs étudiants en ayant de l'information en temps réel sur les questions ou les besoins des étudiants. Les cas d'utilisation de notre application sont :

- Se connecter à la maison. Les étudiants auront la possibilité de se connecter au serveur du cours auquel ils sont inscrits automatiquement dès qu'ils arrivent chez eux. Ainsi ils pourraient automatiquement avoir accès aux dernières informations mises à jour sur le site du cours ou recevoir de l'information concernant les échéanciers des travaux à faire, ou avoir le sujet des cours auxquels ils devront assister le lendemain. Pour le professeur, il pourrait avoir, dès son arrivée à son domicile, un message pour l'informer des différentes questions posées par ses étudiants et un rappel du prochain examen et du sujet du prochain cours qu'il aura à donner.
- Se connecter à l'université. À l'université, l'étudiant pourrait recevoir son emploi du temps de la journée, la liste des travaux à rendre dans la journée, les salles où se dérouleront ses cours, ainsi que les disponibilités de ses professeurs. Le professeur, quant à lui, pourra avoir aussi accès à son emploi du temps de la journée ainsi que le nombre de rendez vous sollicité par ses étudiants.

- Clavarder avec les autres étudiants. Notre architecture permet à un étudiant de converser ou d'envoyer, sans demander une authentification, un fichier à un autre étudiant dès qu'ils se trouvent à proximité. De plus, s'il s'agit d'un professeur et d'un étudiant, ce dernier pourra automatiquement envoyer une question à son professeur dès qu'ils sont à proximité.
- Afficher un document provenant du web. Notre application permet à un étudiant mobile ou un professeur de télécharger un fichier directement du serveur de cours. Cette capacité de notre application à faire des téléchargements du web permettrait à un étudiant qui rentrerait pour la première fois dans un cours de recevoir automatiquement le plan de son cours, sans pour autant devoir aller avec un navigateur sur la page web du cours pour télécharger cette information.
- Mettre à jour la documentation web. La mise à jour de la documentation sur le serveur du cours devra être effectuée par le professeur en charge du cours. Celui-ci aura à mettre à jour les nouvelles concernant son cours, et à organiser la documentation dans le serveur du cours soit manuellement soit à l'aide d'un portail qui va se charger de transformer l'information à envoyer dans un format éditible par un appareil mobile. Le réglage de l'affichage de l'information envoyée est géré par notre application dans l'appareil mobile. Nous prenons ainsi en compte les caractéristiques d'affichage de l'appareil mobile.

3.3.2 Modèle de configuration

Dans cette section, nous proposons une architecture avancée supportant l'adaptation de notre application par rapport au contexte géographique d'utilisation. Nous avons opté pour un modèle client/serveur où le serveur est en attente d'une connexion et le client à la recherche d'un ou de plusieurs serveurs pour se connecter.

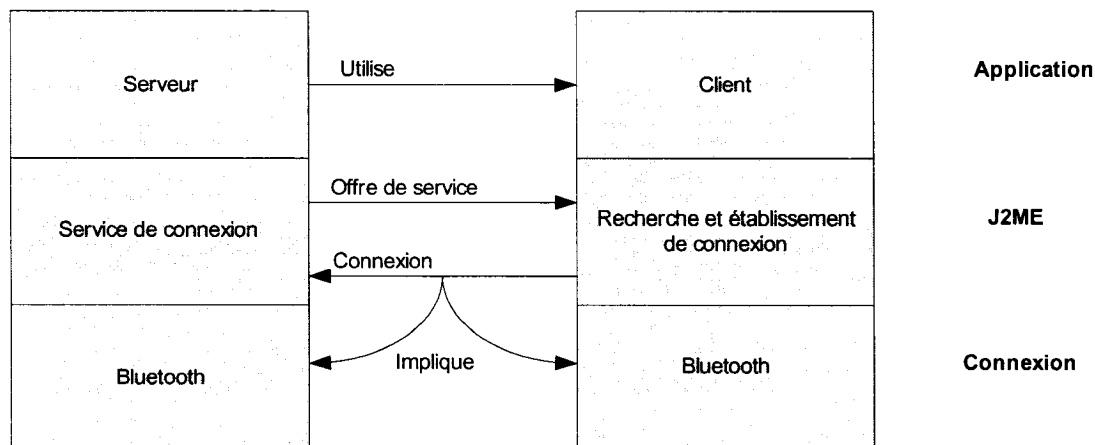


Figure 3.3 Mode client/serveur

À la Figure 3.3, nous présentons les différents niveaux de l'architecture de notre client et de notre serveur. En utilisant la plate-forme J2ME, nous assurons la portabilité de notre application et Bluetooth nous permet la détection du contexte d'utilisation, en plus de servir de support de connexion. Le serveur peut représenter un lieu géographique, un professeur ou un étudiant auquel un ou des clients représentant des étudiants ou un professeur peuvent se connecter.

Supposons le cas où plusieurs étudiants sont dans un même lieu géographique et configurent leur application en mode serveur. Lorsqu'un nouvel étudiant ou le professeur se présente, notre architecture lui permet d'envoyer un message ou un fichier à tous les autres étudiants se trouvant dans ce lieu géographique ou bien de recevoir un message ou un fichier réciproquement. Nous pourrions penser à une salle de classe où le professeur envoie automatiquement les références de son cours à tous les étudiants qui ont un appareil mobile et se trouvant dans la salle.

Création des connexions

Plusieurs serveurs peuvent être exécutés dans des appareils mobiles différents. Chaque serveur attend une connexion d'un client. Ce dernier a été conçu de manière à rechercher en boucle des serveurs se trouvant à sa proximité et fournissant des services. Lorsque le client détecte des serveurs et qu'ils s'identifient, la connexion entre le client

et les serveurs est établie. Cette étape complétée, le client peut envoyer soit un fichier ou un message à tous les serveurs ou recevoir un message ou un fichier de chaque serveur, auquel il est connecté, tel que montré à la Figure 3.4.

Si un serveur va hors de portée du client, sa connexion avec le client est coupée, mais ce dernier reste connecté aux autres serveurs. Cependant, si le client va hors de portée des serveurs, il n'y a plus de connexion établie entre les différents serveurs.

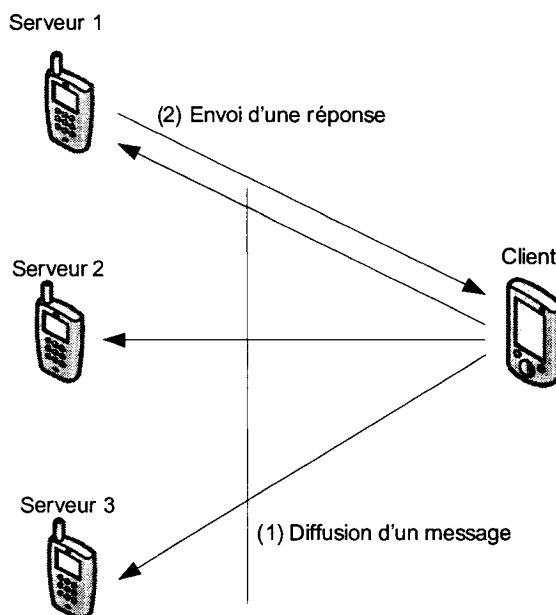


Figure 3.4 Exemple de connexion

Fonctionnalités du serveur

Le serveur a deux interfaces de connexion possibles, comme nous le voyons à la Figure 3.5.

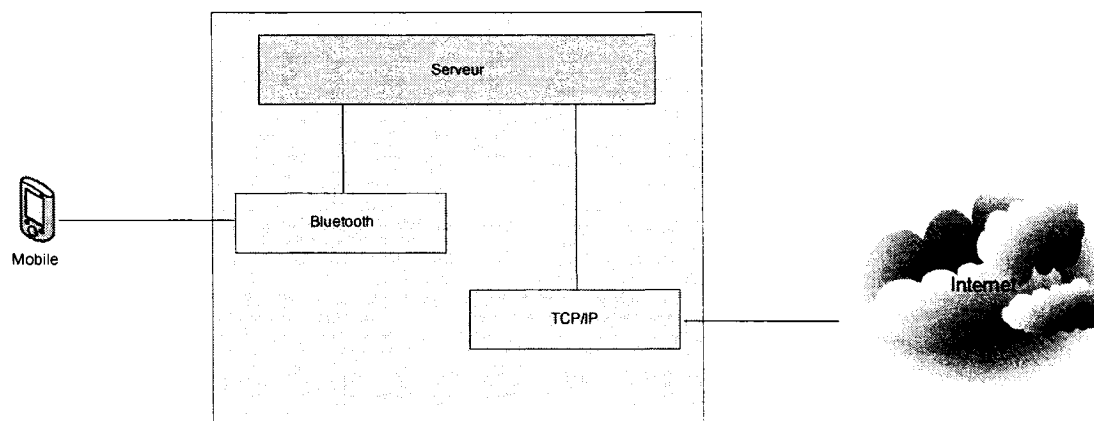


Figure 3.5 Connexions du serveur

En effet, celui-ci se connecte aux clients en utilisant une connexion Bluetooth, mais permet aussi une connexion à un serveur web se trouvant sur Internet. Cette dernière connexion permet à notre serveur de télécharger des fichiers de l'Internet et de les envoyer aux clients connectés. Cette implémentation permet ainsi à notre serveur d'être utilisé comme un relais de communication, et d'éviter ainsi des coûts de connexion élevés pour les étudiants. Dans notre exemple d'implémentation, nous avons créé deux répertoires dans un serveur web qui contiendront dans l'un les fichiers destinés aux étudiants et dans l'autre ceux destinés au professeur.

Lorsque le serveur est configuré comme un point de connexion fixe, et qu'il est connecté à Internet à travers le réseau du fournisseur de service de téléphonie, dès qu'un étudiant entre dans la zone de couverture de son antenne Bluetooth, les deux appareils s'identifient et se connectent. Une fois la connexion établie et l'utilisateur identifié, le serveur mobile n'aura qu'à se connecter à Internet afin de télécharger le fichier adéquat qu'il envoie au client. En fonction du type de fichier reçu, le client utilise l'éditeur approprié de la plate-forme J2ME pour afficher le contenu à l'utilisateur.

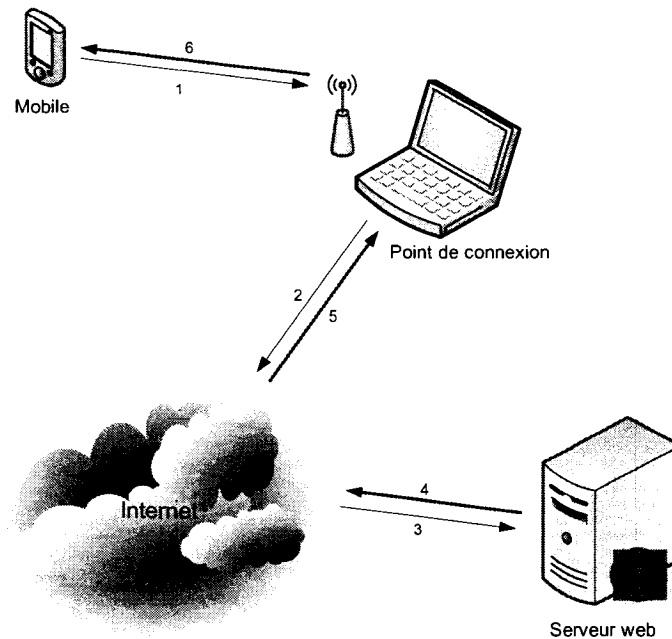


Figure 3.6 Téléchargement d'un fichier d'Internet

3.3.3 Architecture du système de localisation des étudiants mobiles

Les applications clientes peuvent être configurées par l'utilisateur de l'application pour rechercher un serveur spécifique ou se connecter au premier serveur trouvé. Dans le premier cas, l'application cliente se sert de l'identification qui lui est transmise pour détecter le serveur qu'il recherche. Dans le second cas, dès qu'un serveur est trouvé, un fichier ou du texte peuvent être échangés et le client se déconnecte et recommence sa recherche une fois le transfert de données effectué.

Scénari d'utilisation

Le client et le serveur s'identifient à l'aide d'un identificateur fourni par l'utilisateur. De ce fait, le serveur de la maison peut être appelé *home* et celui de la salle de classe, peut être appelée *classroom*. Les étudiants, quant à eux, peuvent être identifiés par leur numéro d'étudiant. En utilisant un identificateur, dès que le client se connecte à un système distant, il demande son identificateur et fournit le sien. Il peut ainsi adapter son fonctionnement dépendamment de l'identificateur qu'il a reçu, qui peut être soit une

personne soit un lieu géographique. Afin de détecter le contexte, chaque utilisateur doit avoir un terminal avec une antenne Bluetooth installée. Ce terminal est soit un PDA, soit un téléphone cellulaire doté d'une antenne Bluetooth.

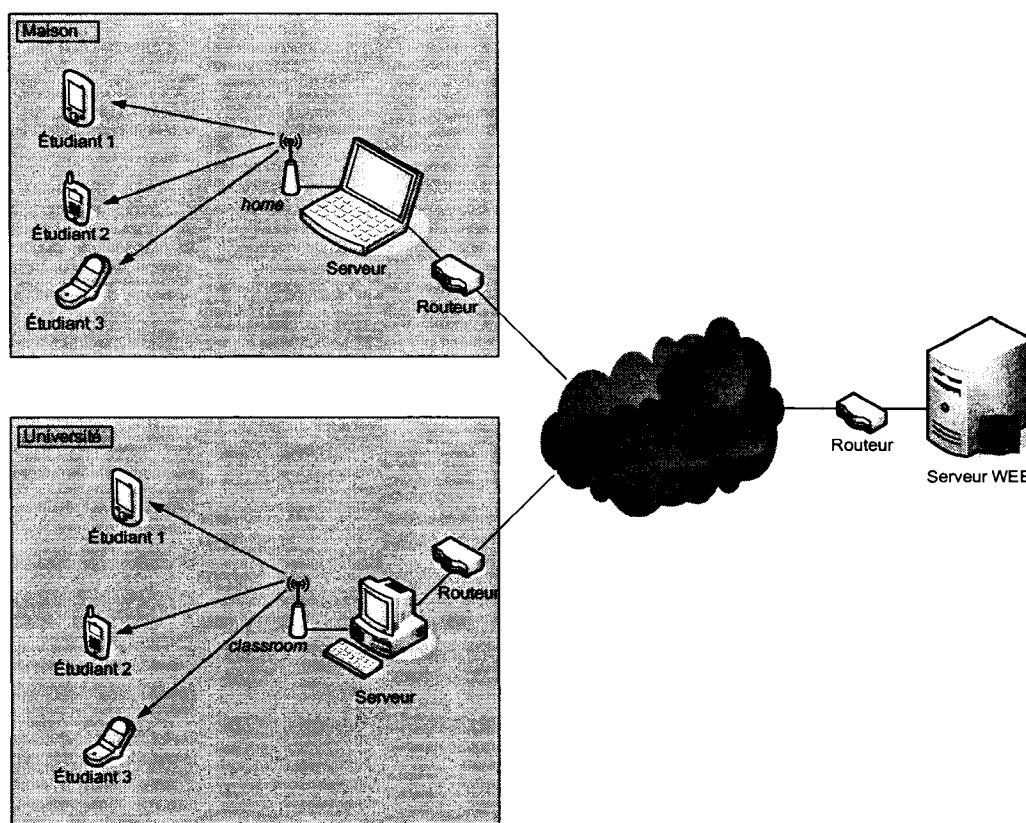


Figure 3.7 Adaptabilité géographique

La Figure 3.7 montre deux serveurs se trouvant à l'université et à la maison. Dès que l'étudiant rentre chez lui, il se connecte automatiquement au serveur de la maison qu'il reconnaît par son identificateur et peut ainsi automatiquement télécharger l'information nécessaire. Le même scénario va se répéter dès qu'il rentre à l'Université où le serveur de l'université lui fournit également l'information dont il a besoin après son identification. Notre application offre aussi la possibilité aux étudiants de se connecter à Internet à travers un autre appareil mobile.

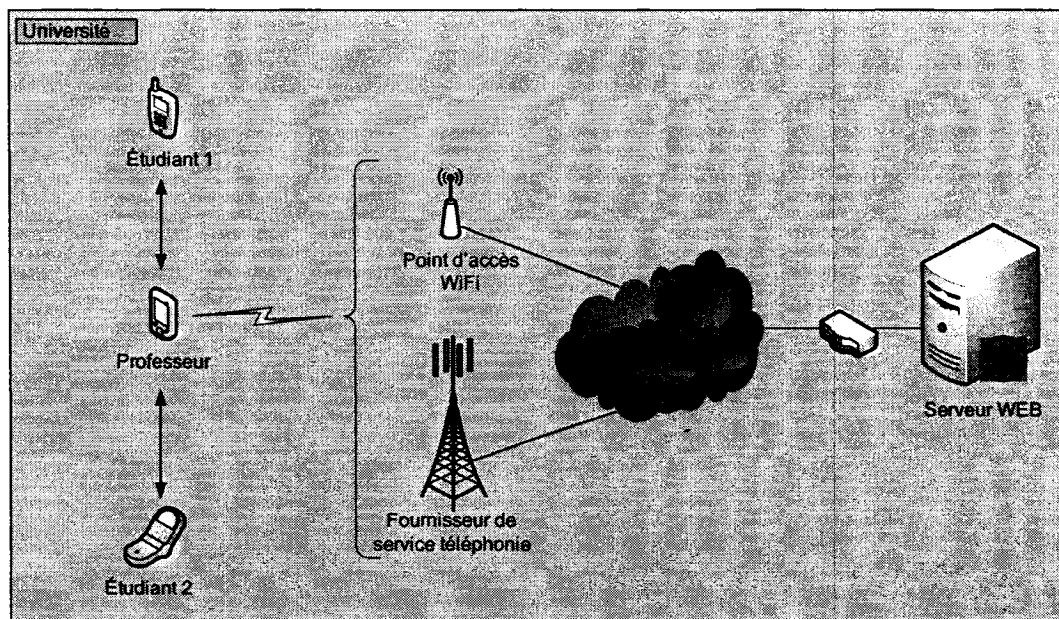


Figure 3.8 Accès aux services sans fil

Dans l'architecture présentée à la Figure 3.8, dès qu'un étudiant s'approche de son professeur, il se connecte à l'application qui roule en mode serveur dans l'appareil mobile de celui-ci, et peut clavier ou recevoir par exemple les disponibilités du professeur pour une séance de consultation, ou les dernières informations concernant le cours et provenant du Web. Cette connexion à internet peut être faite soit par le terminal Bluetooth connecté à internet soit par le fournisseur de service de téléphonie.

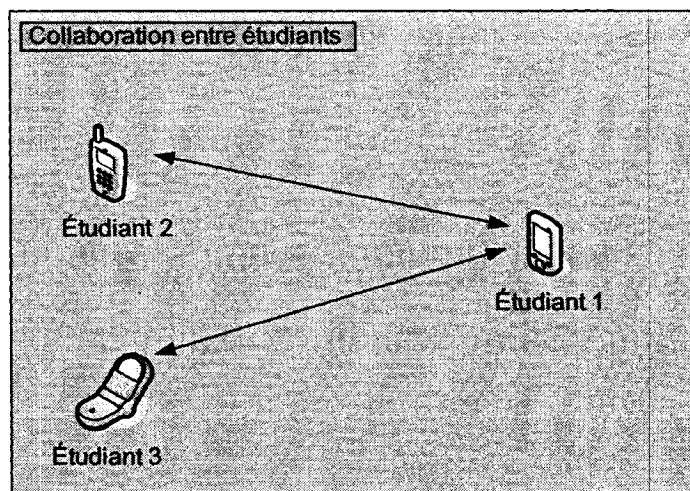


Figure 3.9 Communication multicast entre plusieurs étudiants

La Figure 3.9, montre le scénario de collaboration entre étudiants ou des séances de clavardage sur un thème qui peuvent être organisées dès que les étudiants d'un même cours partagent le même lieu géographique. Le protocole Bluetooth permet aussi l'échange de fichiers devant être partagés dès que des étudiants sont à proximité l'un de l'autre.

Il est important de noter ici que nous avons illustré notre architecture avec des scénarii simples. En effet, nous aurions pu illustrer des architectures beaucoup plus complexes. Cependant, ce type d'architecture ne présente pas un grand intérêt étant donné que le principe est toujours le même, ce qui justifie le choix de simples scénarii. Ceux choisis illustrent bien comment une application, s'exécutant sur une plate-forme mobile, peut s'adapter à son utilisateur, aux caractéristiques de sa plate-forme d'exécution, et à son lieu géographique d'exécution.

En résumé, nous avons analysé les contraintes et les requis d'une application d'apprentissage mobile adaptable. Nous avons ensuite décrit les technologies existantes. Nous avons justifié la technologie utilisée pour implémenter notre modèle d'application d'apprentissage mobile adaptable. Cependant, des tests sont nécessaires, pour prouver le

bon fonctionnement de notre application après implémentation de son architecture. Ceci sera l'objet du prochain chapitre.

CHAPITRE IV

IMPLÉMENTATION, MISE EN ŒUVRE ET RÉSULTATS

Après avoir élaboré une architecture supportant l'adaptabilité des applications d'apprentissage mobile en fonction de leur contexte d'utilisation, une évaluation de son implémentation devrait nous permettre de faire sortir ses avantages. Pour ce faire, nous allons implémenter un prototype de notre architecture supportant tous nos scénari d'utilisation afin d'en évaluer la performance. Dans ce chapitre, nous présentons, dans un premier temps, l'implémentation des différentes unités de l'architecture ainsi que les différents modules utilisés pour gérer l'adaptabilité des applications d'apprentissage. Ensuite, nous présentons l'évaluation de performance en situant d'abord le contexte expérimental, puis en identifiant les indices de performance. Enfin, nous présentons puis analysons les résultats de mesure.

4.1 Environnement d'implémentation

Dans cette section, nous expliquons l'implémentation de nos choix d'architecture pour la mise en œuvre d'un prototype d'application d'apprentissage mobile, capable de s'adapter à son environnement d'exécution ainsi qu'à son environnement géographique. Nous présentons d'abord l'environnement de développement matériel de notre application. Puis nous traitons des caractéristiques de l'environnement logiciel d'implémentation de notre architecture.

4.1.1 Environnement d'exécution matériel

Nous avons réalisé des tests sur notre application pour évaluer les requis qui font intervenir des mesures de performance. Notre banc de test est composé d'un ordinateur portable ayant un processeur Pentium M 1.60 GHz et 528 Moctets de mémoire vive.

Nous avons utilisé le système d'exploitation Windows XP Pro SP 2. Nous allons mener nos expériences à l'aide de plusieurs émulateurs de test fournis avec l'environnement de développement NetBeans. Durant ces tests, le débit de la connexion Internet était de 33 Koctets / seconde en moyenne. Afin de démontrer la portabilité de notre application, nous allons aussi exécuter notre application sur un téléphone cellulaire. Nous avons choisi d'utiliser le modèle 7610 de Nokia (Nokia). Ce dernier est équipé d'une antenne Bluetooth et est fourni avec une plate-forme Java. Ce téléphone a 8 Moctets de mémoire, extensible à 64 Moctets.

4.1.2 Environnement d'exécution logiciel

Notre prototype est exécuté sur un seul ordinateur dans lequel est installé Windows XP. L'application utilise la plate-forme NetBeans (Netbeans) qui s'exécute en utilisant la machine virtuelle de Java JDK1.4 (Java). NetBeans est un outil qui permet aux concepteurs logiciels d'avoir un environnement leur permettant de se concentrer sur leurs programmes. La plate-forme NetBeans est une fondation modulable et extensible utilisée comme plate-forme logicielle pour la création d'applications bureautiques. Les partenaires privilégiés fournissent des modules à valeurs rajoutées qui s'intègrent facilement à la plate-forme et peuvent être utilisés pour développer des outils et solutions. De plus, NetBeans est une plate-forme qui permet le développement d'applications réelles, qui peuvent être émulées dans l'environnement de développement intégré ou déployées sur des appareils mobiles. Nous avons aussi expérimenté Eclipse (Eclipse) qui est une autre plate-forme basée sur Java. Mais il s'est avéré que NetBeans est plus facile d'utilisation et plus spécialisé que Eclipse pour les applications mobiles basées sur J2ME.

L'architecture logicielle offerte par la plate-forme Java est composée de plusieurs modules dont le Mobile Information Device Profile (MIDP) qui est prévu pour les appareils pouvant se connecter à des réseaux, et qui sont caractérisés par des ressources telles qu'un CPU, une mémoire, un clavier, et un espace d'affichage limités (Midp). Le MIDP est un groupe d'API Java qui, avec le Connected Limited Device

(CLD), fournit un environnement d'exécution complet pour les Mobile Information Device (MID). Les spécifications du MIDP JSR-000118 (Jsr118) définissent les requis matériel, logiciel, réseau, ainsi que les API des systèmes standard des appareils qui supportent le MIDP.

Un *Midlet* est un programme Java pour les appareils sans fil, et plus spécifiquement pour les machines virtuelles J2ME. Il s'agit la plupart du temps de jeux pour les téléphones cellulaires. Les *Midlet* s'exécutent sur toutes les plates-formes logicielles supportant le J2ME MIDP. Comme tous les programmes Java, ils n'ont pas besoin d'être compilés pour chaque plate-forme. Pour qu'un *Midlet* puisse s'exécuter sur un téléphone mobile, il faut que :

- sa classe principale hérite de *javax.microedition.midlet.Midlet* ;
- il soit embarqué dans un fichier d'extension .jar ;
- le fichier .jar doit être pré vérifié à l'aide d'un *preverifier*.

Les *Midlet* sont conçus pour coexister avec d'autres applications dans un appareil mobile. Afin de préserver la puissance de calcul des appareils mobiles, les *Midlet* sont implémentés de manière à pouvoir être dans plusieurs états. Ceux-ci permettent aux concepteurs d'applications de prévoir un cycle d'exécution. En général, les *Midlet* peuvent être à l'état pause, activé ou détruit. L'Application Management Software (AMS) permet le contrôle de l'état des *Midlet*. Il construit le *Midlet* et lui communique les changements de son état à l'aide des fonctions suivantes, que chaque *Midlet* doit implémenter : *startApp()*, *pauseApp()*, et *destroyApp()*. Tous les *Midlet* sont à l'état pause au démarrage. Quand un utilisateur démarre une application, le AMS exécute sa fonction *startApp()*, après quoi le *Midlet* est à son état activé, ce qui veut dire que l'interface graphique de l'application s'affiche à l'écran.

4.2 Sous-ensemble de l'architecture implémentée et prototype

Dans cette partie, nous allons discuter de chaque unité et module de notre architecture que nous avons choisi d'implémenter à des fins de test. Pour ce faire, nous

allons utiliser des diagrammes de classe, respectant les spécifications de l'UML (Uml), avec leurs principaux attributs et principales méthodes. L'architecture logicielle de notre prototype d'application est présentée à la Figure 4.1. Nous avons choisi d'implémenter tous les modules logiciels de notre architecture. En effet, ceux-ci sont moins coûteux et nous pouvons facilement les tester avec un environnement d'exécution purement logiciel.

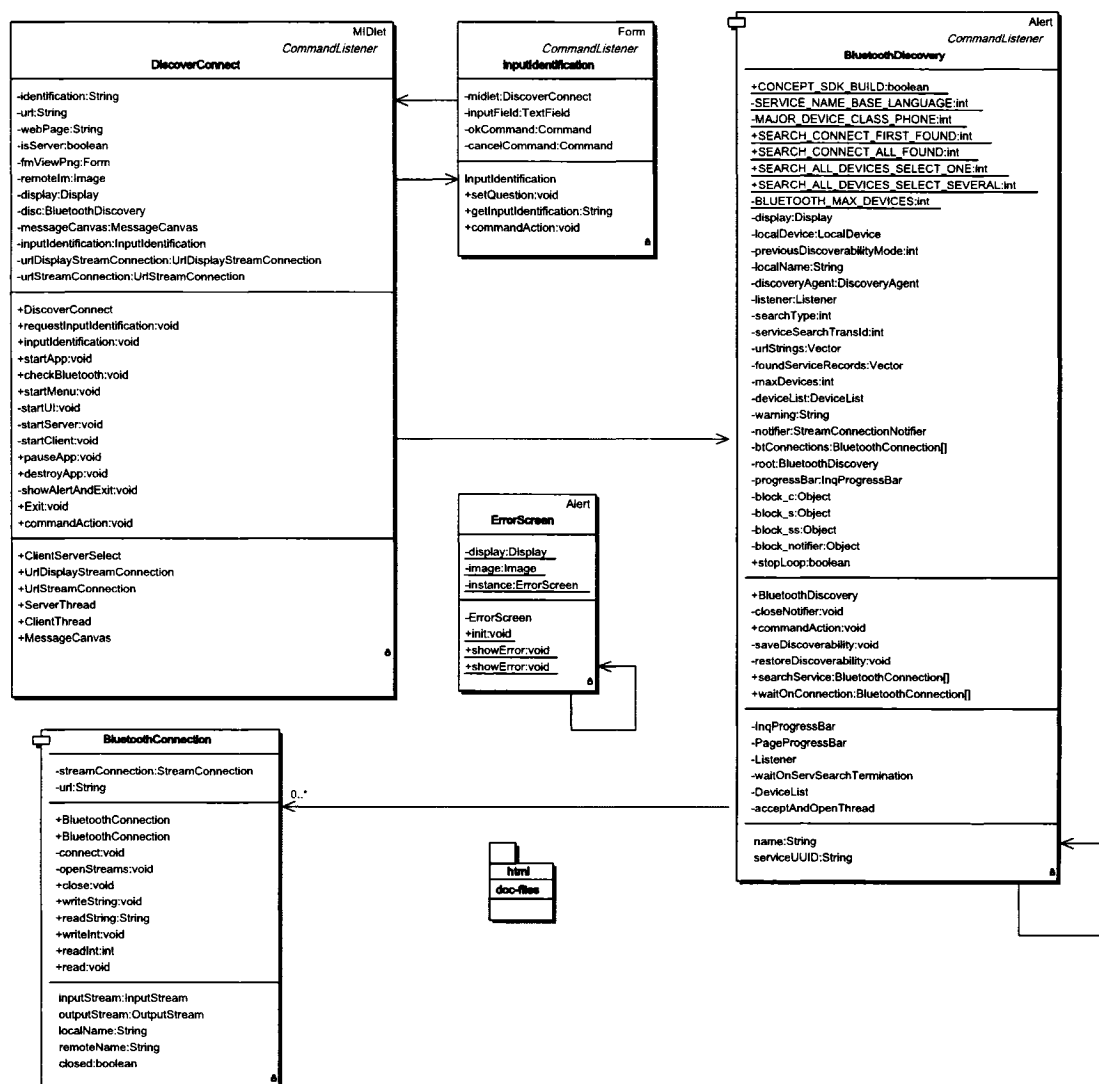


Figure 4.1 Diagramme de classe du prototype

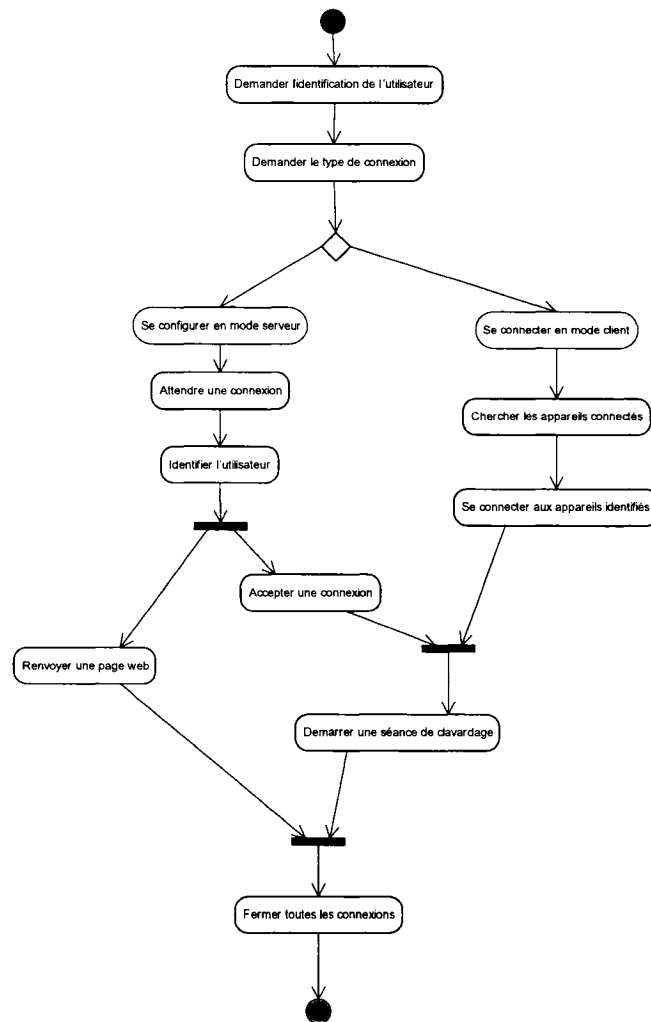


Figure 4.2 Diagramme d'activité du prototype

Notre prototype sera utilisé pour réaliser des expériences afin d'évaluer la performance de notre architecture. L'implémentation que nous avons effectuée démontre la faisabilité de notre modèle en tenant compte des requis et des contraintes que nous avons exposés dans le chapitre 3. Dans notre première configuration, dès que l'application démarre, l'interface graphique demande à l'utilisateur de s'identifier comme montré à la Figure 4.2. Il existe deux types d'utilisateurs, à savoir les utilisateurs mobiles et les stations de base. Pour les stations de base, l'application est en attente d'une connexion. Les applications mobiles quant à eux recherchent en boucle les

stations de base. Dès que la station de base détecte un appareil, une connexion est établie. La station de base se connecte alors à Internet et, en fonction de l'utilisateur mobile, et de la relation qui existe entre l'utilisateur identifié et le lieu de connexion, un fichier est téléchargé sur la station de base. Ce fichier est ensuite transféré à l'appareil mobile via leur connexion Bluetooth.

Identification des utilisateurs

La classe *DiscoverConnect* hérite de la classe *Midlet*. Celle-ci est la classe principale. Lorsque l'application est démarrée par l'utilisateur, la classe *InputIdentification* est instanciée et demande, à l'aide d'une interface graphique adéquate, l'identification de l'utilisateur. Cette classe hérite de la classe *Form* qui est en fait une boîte de dialogue sur laquelle peut être dessinée une boîte de texte. L'utilisateur entre son identification sur celui-ci à l'aide du clavier de l'appareil mobile. Cet identificateur peut être un nom d'utilisateur, un numéro s'il s'agit d'un étudiant, un sigle de cours s'il s'agit d'un professeur ou d'un nom de lieu s'il s'agit de point de connexion fixe. Si l'opérateur omet de s'identifier, l'application prend comme identification le nom par défaut de l'appareil mobile. La classe *InputIdentification* est représentée à la Figure 4.3.

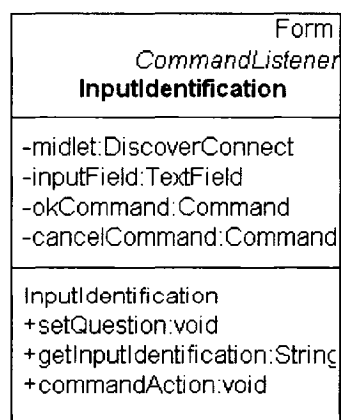


Figure 4.3 Classe InputIdentification

Initialisation de Bluetooth

La première phase de la connexion Bluetooth est la recherche de nouveaux appareils mobiles qui utilisent leur antenne Bluetooth à l'aide de la classe *BluetoothDiscovery* représentée à la Figure 4.4. Cette classe possède comme attribut la classe *DiscoveryAgent*. Cette classe permet de répertorier l'ensemble des appareils Bluetooth dans un espace géographique.

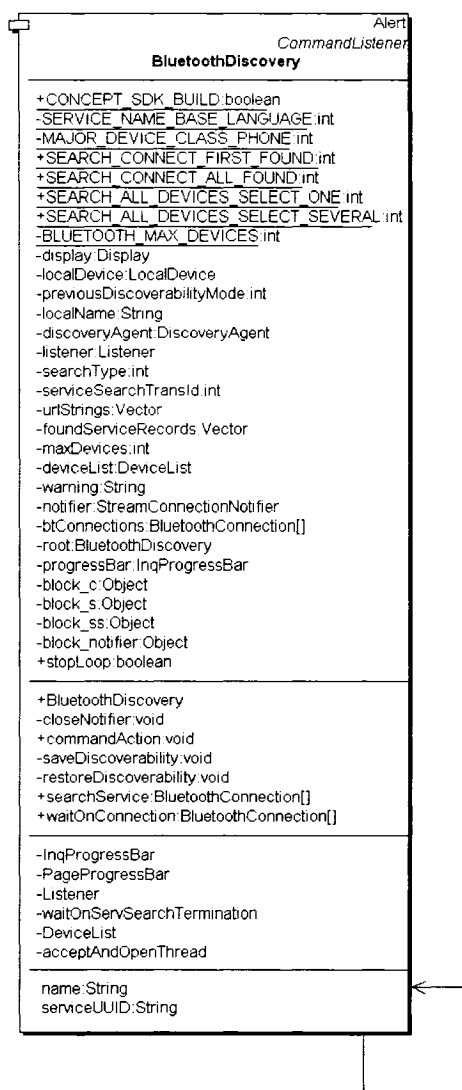


Figure 4.4 Classe *BluetoothDiscovery*

La classe *DiscoveryAgent* fournit les méthodes nécessaires à la détection des appareils mobiles. Cette recherche peut être générale, avec une recherche étendue à tous les appareils se trouvant dans le périmètre ou sélective s'il y a une sélection des connexions basée sur le type (Class of Device) des appareils. La réception de l'identification de l'appareil distant, est effectuée à l'aide de la fonction *getFriendlyName()* de la classe *RemoteDevice*. Notre implémentation du protocole Bluetooth permet de continuer la recherche d'autres appareils à proximité même s'il y a déjà des clients connectés.

Choix du mode d'opération

Une fois que l'utilisateur s'est identifié et la connexion Bluetooth initialisée, la classe *ClientServerSelect* est instanciée. Cette classe hérite de la classe Java *List* et propose à l'utilisateur un choix de mode de connexion. L'utilisateur a le choix d'utiliser son application en mode client ou en mode serveur. Les différents choix qui lui sont proposés sont :

- *Wait Conn ;*
- *Find First Conn ;*
- *Find All Conn ;*
- *Select a Conn ;*
- *Find Several Conn.*

Le premier choix est pour le cas où l'utilisateur de notre application veut l'utiliser comme serveur. Dans ce cas d'utilisation, l'application instancie la classe *ServeurThread*. Cette classe hérite de la classe *Thread*, qui permet d'exécuter simultanément, plusieurs tâches. Cette caractéristique des *thread* nous permet dans notre cas, d'instancier plusieurs fois la classe *ServeurThread* quand plusieurs applications clientes veulent se connecter au même serveur.

Les autres choix concernent le cas où l'utilisateur voudrait utiliser notre application en mode client. Il peut donc décider de se connecter au premier serveur qu'il détecte. Dans ce cas, l'application recherche en boucle un serveur pour se connecter.

L'utilisateur peut aussi décider de se connecter à tous les serveurs se trouvant à la portée de l'antenne Bluetooth. Ou se connecter à un serveur spécifique ou enfin à un groupe de serveurs. Dans ces différents cas, la classe *ClientThread* qui hérite de *Thread* est instanciée.

Connexion entre utilisateurs

Comme nous l'avons vu, les étudiants ou les professeurs peuvent configurer leur application en mode client ou en mode serveur. Lorsque deux utilisateurs mobiles se détectent, la classe *BluetoothConnection* montrée à la Figure 4.5 est instanciée afin d'établir une connexion Bluetooth entre les deux applications.

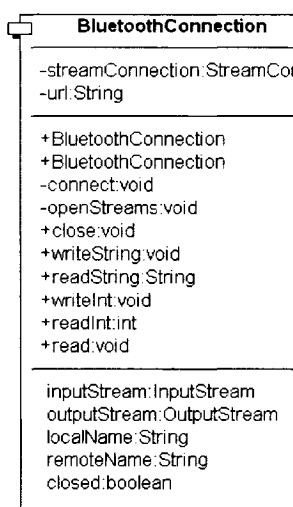


Figure 4.5 Classe BluetoothConnection

Une fois que la connexion est établie, la classe *MessageCanvas* représentée à la Figure 4.6, définit un flux de données entre les deux mobiles. Les deux utilisateurs ont le choix d'utiliser leur clavier afin de se transférer des fichiers ou des messages textuels. Un réseau d'échange de données peut donc s'établir de manière ad hoc, dans un lieu géographique. Nous pourrions penser à une salle de classe ou à un local d'étudiants.

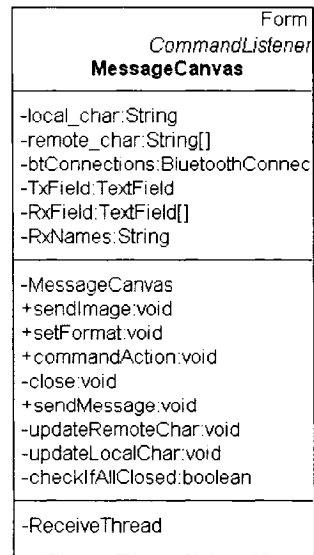


Figure 4.6 Classe MessageCanvas

Détection d'un contexte géographique

Lorsque notre application doit s'exécuter sur une plate-forme qui est identifiée comme un lieu géographique, il doit être configuré en mode serveur. Dans ce cas, dès qu'un nouvel utilisateur est détecté, le serveur instancie la classe *UrlStreamConnection* représentée à la Figure 4.7, afin d'établir une connexion à un serveur Web à travers Internet. Ce qui lui permet de télécharger un fichier se trouvant dans un dossier, en rapport avec l'identité du client connecté et le lieu de connexion.

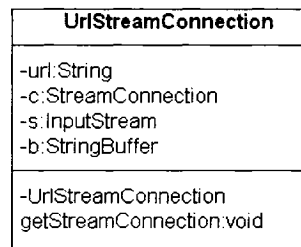


Figure 4.7 Classe UrlStreamConnection

Une fois que le fichier est téléchargé du Web, il est transmis au nouveau client connecté. Ce fichier peut être un fichier texte ou un fichier image de type ‘‘png’’, que l'utilisateur pourra visualiser à l'aide de l'écran de son appareil mobile.

4.3 Analyse de performance

Maintenant que nous avons développé un prototype qui implémente les différentes fonctionnalités de notre modèle, nous effectuons des mesures de performance sur notre prototype. Ces mesures serviront à évaluer la faisabilité et la performance de notre modèle. L'objectif d'une telle analyse est de déterminer le degré d'efficacité avec lequel les ressources utilisées sont gérées. Dans cette partie, nous présentons au préalable le banc de test et l'environnement expérimental que nous avons utilisé. Puis, nous exposons et analysons les résultats des mesures que nous avons effectuées. Finalement, nous discutons puis évaluons nos résultats.

4.3.1 Choix des indices de performance

Le problème dans cette analyse est de savoir comment notre modèle permet à des utilisateurs de se détecter, de former des réseaux ad hoc et de s'envoyer des fichiers tout en étant mobiles. L'objectif est donc de détecter le contexte d'utilisation et de mesurer le temps de transfert des données entre utilisateurs. Nous définissons comme temps de connexion, le délai de transfert du fichier auquel on ajoute le délai requis pour la phase d'identification des utilisateurs entre les deux appareils.

Nous avons le choix parmi plusieurs indices de performance pour décider de la possibilité d'implémenter notre modèle d'application. Parmi ceux que nous avons identifiés, il y a le temps de réponse par rapport aux appareils, la fiabilité et la robustesse de notre modèle, ainsi que la disponibilité des différents moyens de connexion. Tous ces facteurs de performance ne sont pas applicables car nos tests sont effectués dans un environnement de test logiciel. Nous utilisons donc comme principal indice de performance le temps requis pour la détection et le transfert des données entre utilisateurs.

Délai de transfert de fichier

Le délai de transfert de fichier mesure le temps qu'il faut à deux applications devant s'envoyer un fichier. Comme nous l'avons vu dans le chapitre 3, notre application configurée en mode serveur peut être connectée à Internet afin de télécharger les données d'un serveur web, pour les transférer au client à l'aide de sa connexion Bluetooth. Pour évaluer notre modèle, nous allons effectuer trois expériences. Prenons le cas où un utilisateur est en mouvement et pénètre dans la zone de couverture d'une application serveur. Ce dernier n'a que très peu de temps pour détecter le client et lui transférer les données appropriées.

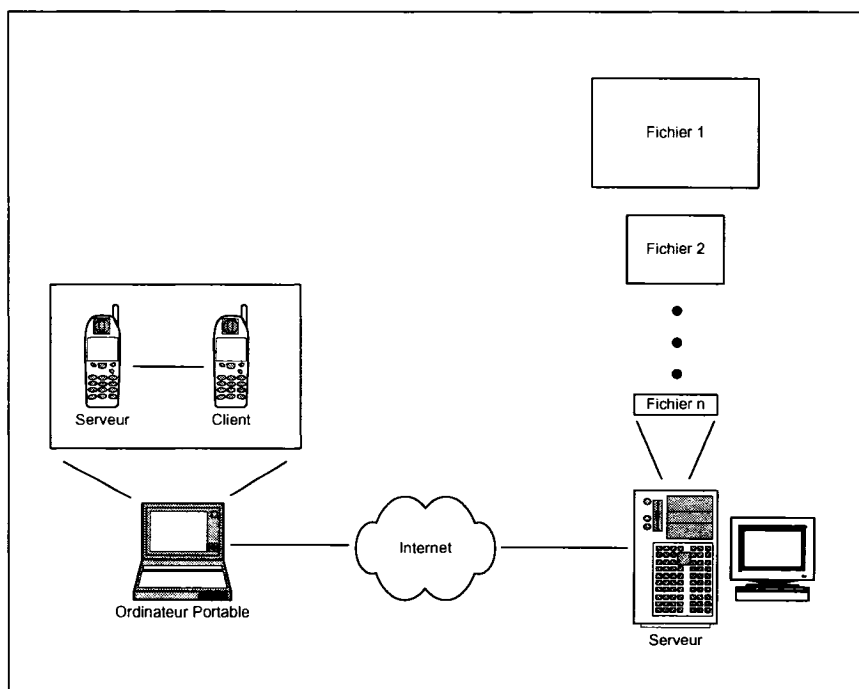


Figure 4.8 Impact de la taille des fichiers sur le délai de transfert

Nous allons par la première expérience illustrée à la Figure 4.8, chercher l'impact de la taille des fichiers téléchargés d'Internet, et envoyés au client, sur le temps de transfert du fichier requis entre les deux appareils. Nous allons ainsi simuler le fait qu'un

utilisateur mobile traverse une zone couverte par une antenne Bluetooth, afin de connaître le temps que celui-ci doit rester dans la zone de couverture afin de recevoir l'ensemble de l'information qui lui est destiné. Soit TA réception, le temps requis afin de recevoir le fichier envoyé par le serveur pour l'application cliente, TA Bluetooth le temps nécessaire pour transférer le fichier du serveur au client à l'aide de la connexion Bluetooth, TA Internet le temps nécessaire au serveur pour télécharger un fichier d'Internet, et TA Calcul le temps nécessaire au serveur mobile pour déterminer le fichier adéquat à envoyer au client connecté. Nous pouvons dire que :

$$TA \text{ réception} = TA \text{ Bluetooth} + TA \text{ Internet} + TA \text{ Calcul}$$

En mesurant le temps de transfert, nous serons en mesure de déterminer le temps requis pour la connexion au serveur et pour le téléchargement de l'information dont il a besoin, en fonction de la quantité d'information qui lui est destinée. Nous n'allons pas considérer TA Calcul dans nos mesures car le délai requis pour cette opération est très variable et est en fonction de la technologie utilisée pour l'implémenter.

Notre deuxième expérience, consiste à considérer un client qui se connecte à plusieurs serveurs qui lui envoient de l'information provenant d'Internet. Nous allons avec cette architecture de test présentée à la Figure 4.9, mesurer le temps de réception des fichiers par le client, en fonction du nombre de serveurs qui transmettent de l'information. Nous allons par cette expérience, simuler un client qui pénètre dans une zone couverte par plusieurs serveurs auxquels il se connecte afin de recueillir des données qui lui sont destinées.

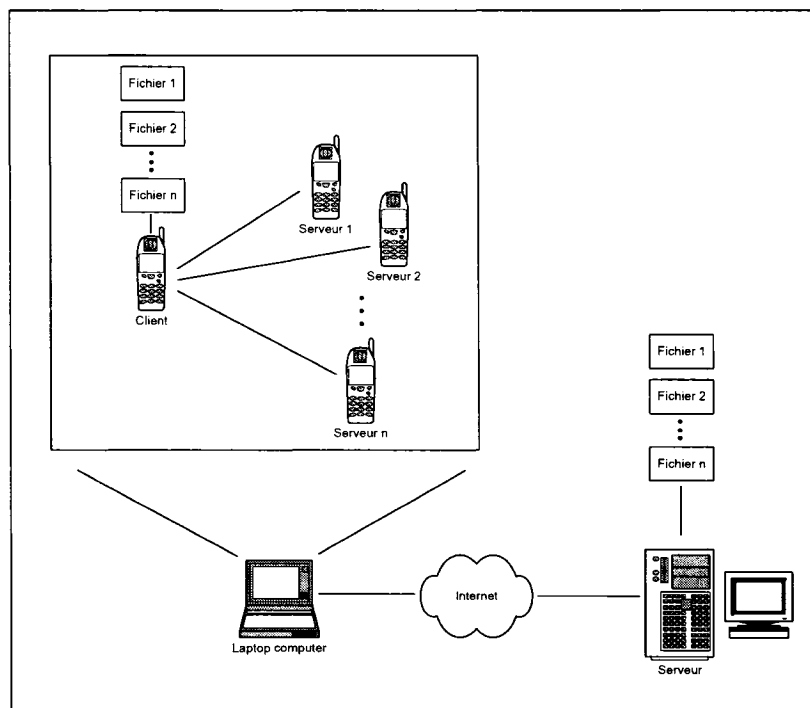


Figure 4.9 Connexion d'un client mobile à plusieurs serveurs

Soit T_1 le temps requis pour télécharger le fichier provenant d'Internet du premier serveur, T_2 celui du deuxième serveur et T_n celui du serveur n . Nous allons déterminer avec cette expérience, le temps requis pour une application mobile, pour recevoir concomitamment de l'information de plusieurs serveurs se trouvant dans un même lieu géographique. Nous avons donc les formules :

$$T_n = T_n \text{ Internet} + T_n \text{ Bluetooth}$$

$$TB \text{ réception} = \text{Max} (T_1, T_2, \dots, T_n)$$

Notre dernière expérience consiste à exécuter dans notre environnement de simulation plusieurs clients qui se connectent à un même serveur. Ce test nous permettra de mesurer le délai requis pour un serveur, d'envoyer des fichiers provenant de l'Internet à plusieurs clients. L'architecture de test est présentée à la Figure 4.10.

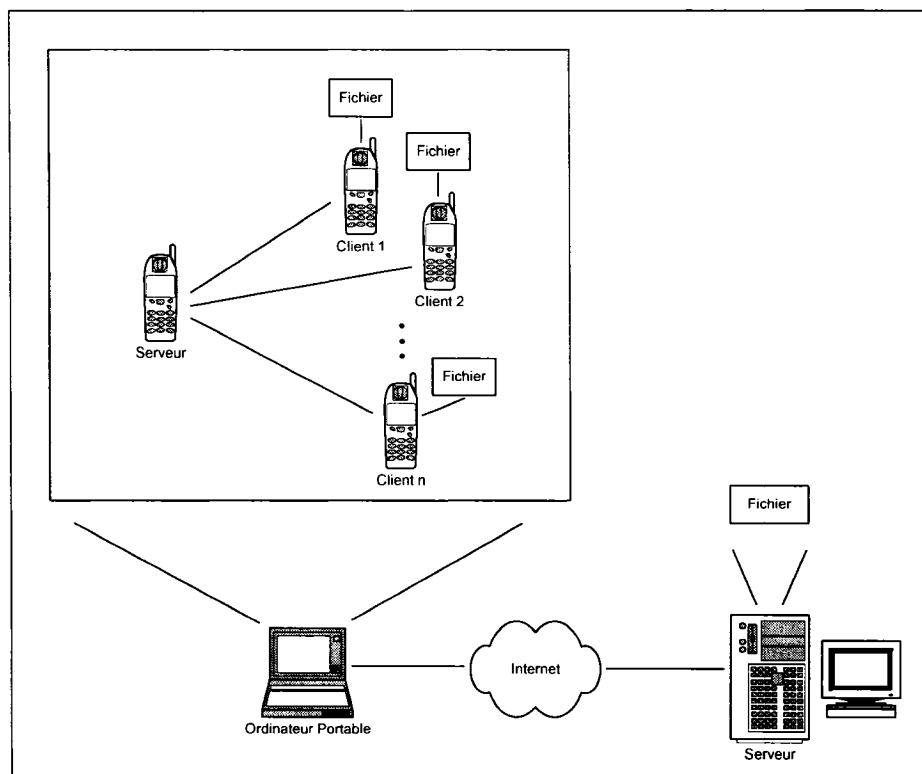


Figure 4.10 Connexion de plusieurs clients à un serveur

Ce test va nous permettre de mesurer la variation du temps requis pour envoyer une page provenant de l'Internet par un serveur, à plusieurs clients qui entrent dans sa zone de couverture. Soit T_1 le délai requis pour que le premier client reçoive l'information qui lui est destinée. Et T_i le temps requis pour que le i ème client reçoive le fichier. Nous avons donc :

$$T_i = T_i \text{ Internet} + T_i \text{ Bluetooth}$$

Le temps de réception mesuré par rapport au nombre de clients, va nous permettre de savoir s'il faut plus de temps à un client pour télécharger l'information qui lui est destinée s'il pénètre dans une zone couverte par un serveur auquel plusieurs clients sont déjà connectés.

Les facteurs externes qui pourraient affecter nos mesures et qui sont applicables à toutes nos expériences sont :

- La vitesse de la connexion Internet : dans notre modèle, les serveurs à cause de leur capacité de stockage limité, téléchargent les informations requises d'Internet. Les fichiers les plus fréquemment envoyés pourraient être stockés dans le serveur, de manière à minimiser le nombre de connexions requis.
- Le temps nécessaire au serveur pour décider de l'information adéquate à télécharger puis à transférer au client : lorsque le serveur détecte le client, l'application doit décider du fichier à envoyer en fonction du contexte. Le temps requis pour cette opération dépend de la complexité du calcul du contexte. Nous allons faire fi de ce facteur, étant donné que notre étude va porter d'avantage sur la vitesse de connexion et de transfert des données requise pour notre modèle.
- La vitesse du processeur de l'appareil mobile utilisé : facteur qui influence le délai pour afficher les informations reçues. Afin de minimiser cet impact, nous n'allons pas prendre en compte lors de notre mesure ce délai. Étant donné qu'il pourrait y avoir une variation significative en fonction de l'appareil utilisé et que nous allons utiliser les mêmes appareils mobiles, seule la variation relative de ce délai est significative pour nos mesures.

Test de portabilité par une utilisation sur plusieurs plate-formes

Afin de vérifier que l'interface usager que nous avons implémentée est portable, et qu'elle est adaptable en fonction de l'espace d'affichage et des boutons de contrôle des appareils mobiles, nous allons exécuter notre application sur plusieurs plates-formes d'appareils différents. Pour ce faire, nous allons utiliser les émulateurs de téléphones disponibles et intégrés à notre environnement de développement pour vérifier les fonctionnalités que nous avons implémentées.

De plus, nous allons exécuter notre application sur un téléphone cellulaire Nokia 7610. Nous pouvons ainsi confirmer que notre application peut bien être exécutée sur un appareil mobile et qu'il prend en considération les caractéristiques de l'appareil.

4.3.2 Réalisation des expériences

Dans cette sous section, nous présentons les mesures que nous avons effectuées pour évaluer notre solution. Les indices de performance dépendent de la taille des fichiers transférés, et du nombre d'appareils mobiles dans le réseau.

La mesure du délai de transfert des données est réalisée à l'aide d'appels à la fonction *System.currentTimeMillis()* de Java. Les appels à cette fonction sont soit effectués au niveau du client, lorsque l'on fait varier la taille du fichier envoyé, ou qu'un client se connecte à plusieurs serveurs. Soit les mesures sont effectuées au niveau du serveur, lorsque plusieurs clients se connectent au même serveur. Lorsque le client doit se connecter à plusieurs serveurs, l'appel de cette fonction est effectué dès que deux appareils se détectent. Ce qui nous permet de mesurer le temps de transfert de fichier entre le client et le serveur. Pour nos mesures, nous prenons en considération le temps de réponse Internet et le temps de transfert induit par la connexion Bluetooth. Pour le client, le temps de transfert implique :

- le délai requis pour la phase d'identification des utilisateurs entre les deux appareils ;
- le temps requis pour établir une connexion entre le serveur sans fil et le serveur Internet sur lequel se trouve le fichier à télécharger ;
- le temps requis pour télécharger le fichier de l'Internet par le serveur et le temps nécessaire au client pour recevoir le fichier du serveur sans fil à l'aide de la connexion Bluetooth.

Pour le serveur, le temps de transfert implique :

- le temps requis pour établir une connexion avec un serveur Internet et pour télécharger un fichier une fois qu'un client se connecte par l'interface Bluetooth ;

- le temps requis pour envoyer le fichier téléchargé d'Internet au client par la connexion Bluetooth.

Les résultats présentés ci-dessous sont obtenus à partir de plusieurs sessions d'évaluation réalisées durant deux semaines. Durant cette période, nous avons utilisé un ordinateur portable avec un processeur de 1.6GHz et une mémoire de 512 Ko qui nous a permis de faire des tests et des mesures sur une même plate-forme Java.

Étant donné certaines mesures de temps étaient relativement instables avec des écarts types pouvant dépasser 10%, nous reportons les données à partir plusieurs sessions de mesures. Nous prenons la moyenne de 5 sessions avec un écart type inférieur à 5%, puis faisons abstraction des autres sessions.

La taille des fichiers

L'influence de la variation de la taille des fichiers transférés au client et provenant d'Internet est présenté à la Figure 4.11. Nous avons fait varier la taille d'un fichier envoyé par le serveur au client, afin de mesurer le temps de transfert requis entre les deux appareils entre le moment où le client détecte le serveur et celui où il reçoit un fichier du serveur. Les mesures rapportées sur le graphique constituent une moyenne obtenue à partir de 5 sessions de test.

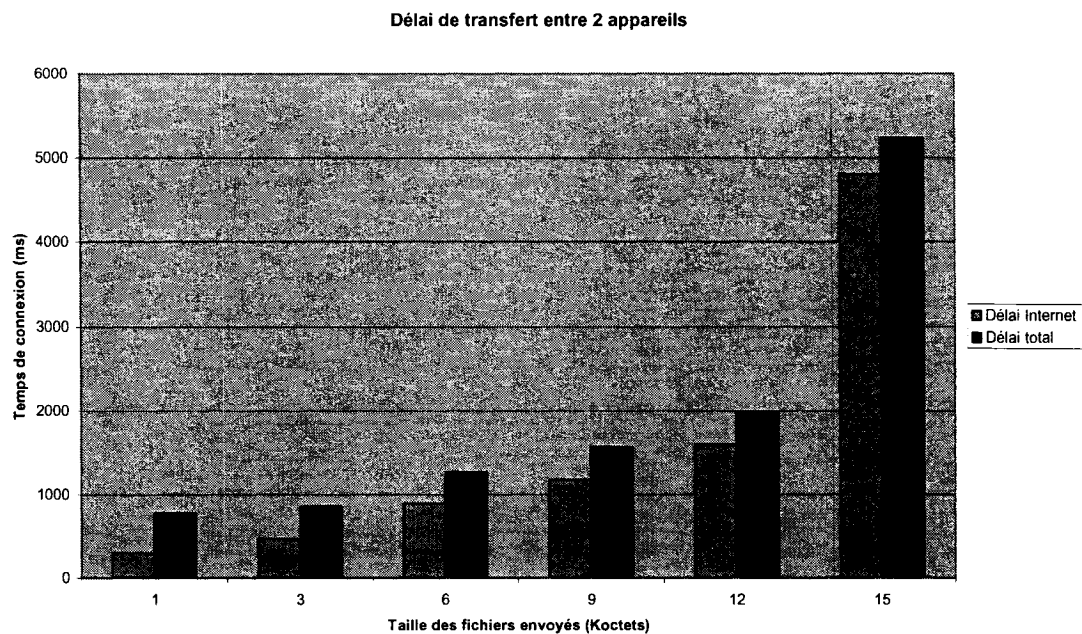


Figure 4.11 Temps de réponse en fonction de la taille des fichiers

Comme nous pouvons le voir à la Figure 4.11, le temps de transfert total requis pour qu'un client reçoive un fichier croît proportionnellement à la taille du fichier envoyé. Pour un fichier de la taille de 1 Koctets, le délai requis pour l'établissement de la communication Bluetooth avec le serveur et la réception du fichier est de moins de 800 ms. Ce qui voudrait dire qu'un client en déplacement pourrait recevoir le fichier sur son appareil, tout en restant en mouvement s'il vient assez proche du serveur. Pour une taille de fichier de 15 Koctets, nous voyons que le délai de transfert total requis est de plus de 5 secondes. Ce temps étant relativement très long, l'envoi d'un fichier de cette taille nécessiterait que le client s'arrête à côté du serveur afin de permettre aux deux applications de s'échanger l'information.

En ce qui concerne le délai de transfert Bluetooth, nous constatons que malgré la variation de la taille des fichiers envoyés, le temps de transfert reste sensiblement le même. En effet, comme nous avons pu le voir dans le chapitre précédent, le débit de transfert maximal de données permis par Bluetooth est de 721 Kbits / seconde, ce qui

correspond à 90 Koctets / seconde. Nous pouvons en déduire que pour un fichier de 15 Koctets, le temps de transfert serait de 166 millisecondes, ce qui est très comparable aux résultats que nous avons obtenus. Pour les fichiers de plus petite taille, nous constatons que le délai de transfert est presque le même que le temps requis aux fichiers de plus grande taille. Ceci peut être expliqué par le fait que nous avons effectué nos tests sur une plate-forme Windows qui n'est pas un système d'exploitation temps réel, et qui offre une erreur relative de 55 millisecondes (Windows). Nous pouvons donc en déduire que le délai induit par le transfert de nos fichiers par le protocole Bluetooth est trop court pour que nous puissions en faire une mesure précise.

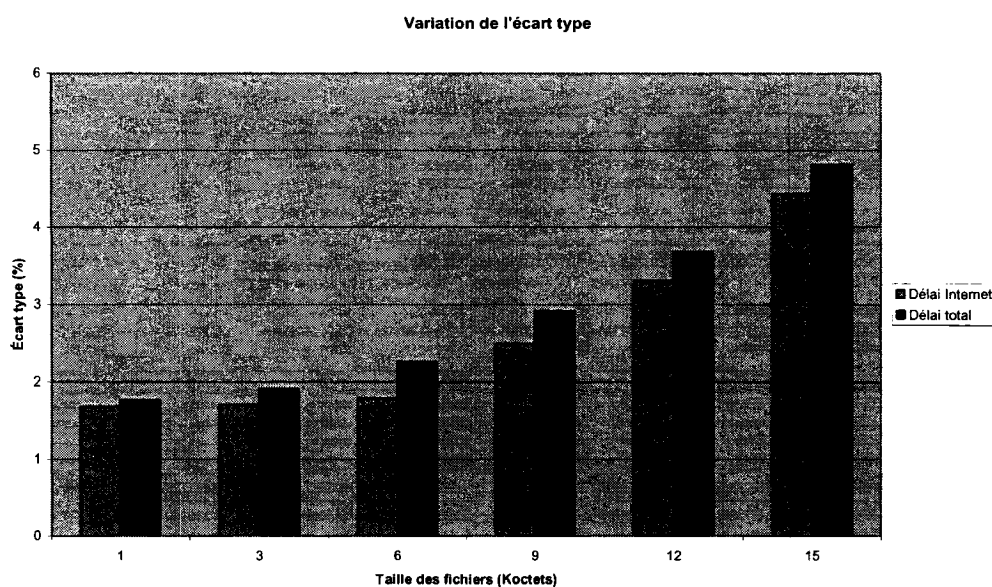


Figure 4.12 Écart type en fonction de la taille des fichiers (N=5)

Dans la figure 4.12, nous calculons l'écart type à partir de 5 données recueillies. Nous nous rendons compte que c'est la mesure du délai Internet qui varie le plus. Ceci est dû à la charge non prédictible que peut avoir un serveur Internet du point de vue d'un utilisateur distant.

À travers cette expérience, nous voyons que dans le cas d'un simple transfert de fichier, c'est la connexion Internet qui constitue le goulot d'étranglement en terme de vitesse de transmission des données. Nous allons limiter nos expériences à des fichiers

de 15 Koctets, étant donné que les appareils mobiles sont, la plupart du temps, limités par leur capacité de mémoire.

Connexion d'un client à plusieurs serveurs (1 K octets)

Voyons maintenant l'influence de la variation du nombre de serveurs sur le temps de transfert requis pour le transfert de fichiers à un client. Pour chaque prise de mesure que nous avons effectuée, nous avons exécuté le nombre de serveurs adéquat. Ensuite, nous avons exécuté un client configuré pour détecter tous les serveurs qui sont à proximité dans un instant donné. Nos mesures sont effectuées à partir des différents serveurs et nous avons relevé le temps maximal de transfert pour aussi bien le délai Internet que le délai de transfert Bluetooth. Nous avons commencé à effectuer nos mesures à partir 5 sessions de mesure.

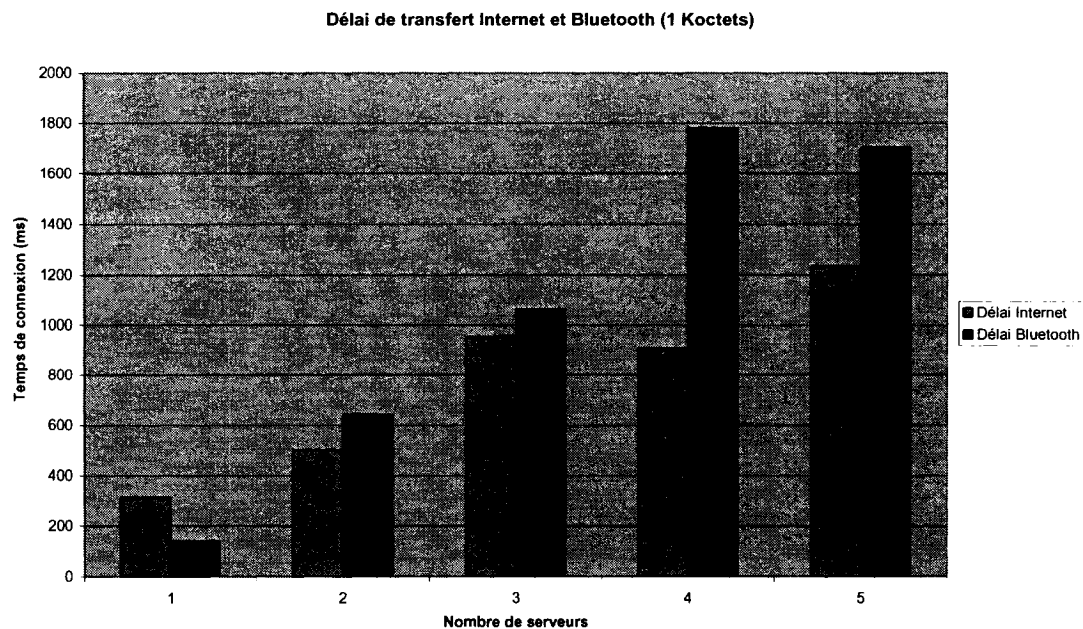


Figure 4.13 Variation du délai de transfert entre un client et plusieurs serveurs

Contrairement à ce que nous pourrions nous attendre suite à notre dernière expérience, nous voyons à la Figure 4.12, que le temps de transfert Bluetooth dépasse rapidement le délai de transfert induit par la connexion Internet. En effet, pour un fichier de 1 Koctets et un serveur, le délai de transfert Internet est de 313 ms et le temps de transfert Bluetooth de 140 ms. Cependant, dès que nous augmentons le nombre de serveurs à 2, nous constatons que le temps de transfert Bluetooth dépasse le temps de transfert Internet. Ceci est dû au fait que les transferts du client aux serveurs sont quasiment instantanés et tous les serveurs mobiles adressent une requête presque en même temps au serveur Internet. De plus ils envoient tous leur fichier au client par la connexion Bluetooth en même temps. Ceci induit une surcharge du lien Bluetooth qui se traduit par une augmentation du temps de transfert. Ainsi pour 5 serveurs, nous avons mesuré un temps de transfert Bluetooth de 2937 ms, ce qui est plus du double du temps de la transmission Internet qui est de 1234 ms.

Nous constatons ici que le nombre de serveurs qui peuvent se trouver dans une zone géographique et auxquels un client se connecte en même temps, a une forte influence sur le temps de transfert de fichier du client aux serveurs.

Connexion d'un client à plusieurs serveurs (15 KOctets)

À la Figure 4.13, nous avons fixé la taille du fichier transféré à 15 Koctets et nous avons effectué la même mesure.

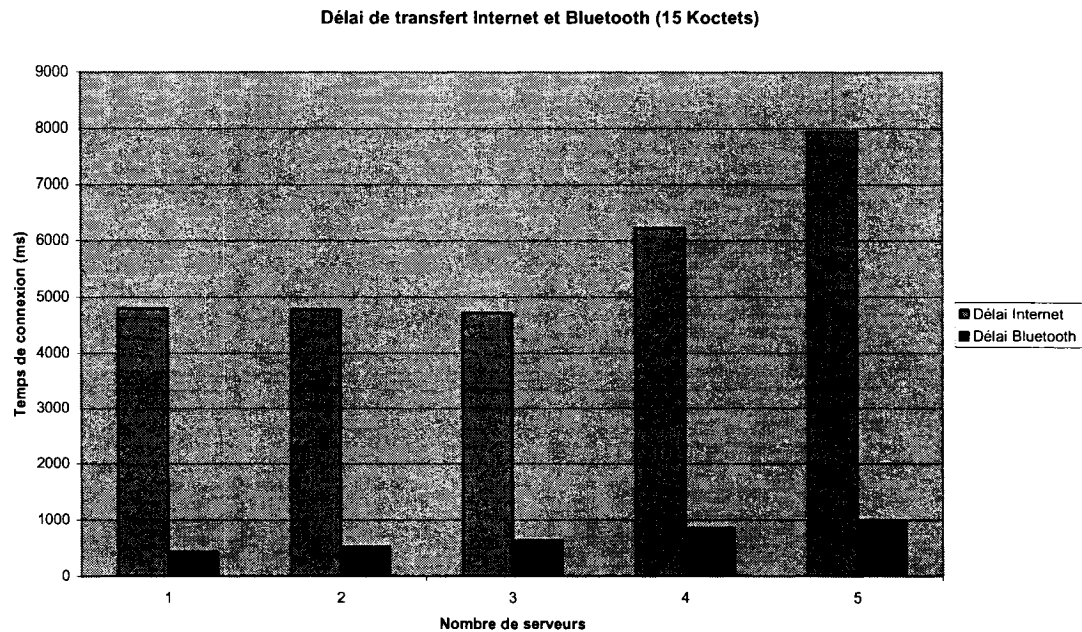


Figure 4.14 Variation du délai de transfert entre un client et plusieurs serveurs

Dans ce cas, nous voyons que le délai de transfert Internet pour un serveur est de 4797 ms et 437 ms pour le temps de transfert Bluetooth avec un fichier de 15 Koctets. Si nous augmentons le nombre de serveurs qui se connectent au même client à 5, nous mesurons un délai de transfert Internet de 7937 ms et un temps de transfert Bluetooth de 1001 ms. Malgré une augmentation du temps de transfert Bluetooth, nous constatons que le délai de transfert Internet est presque 8 fois plus élevé. En faisant une étude comparative des Figures 4.12 et 4.13, nous pouvons déduire que ce qui prend le plus de temps pour les transferts Bluetooth est l'établissement de la communication qui peut être très coûteux. De plus, nous pouvons relever le fait que les connexions Bluetooth sont très efficaces dans le transfert de fichiers de grande taille. En ce qui concerne le temps de transfert global du fichier de 15 Koctets, de la détection des serveurs à la réception des fichiers, nous voyons que celle-ci varie de 5234 ms à 8938 ms. Ce délai de transfert implique qu'un client qui pénètre dans une zone couverte par plusieurs serveurs devant

lui envoyer un fichier nécessite que l'utilisateur mobile s'arrête devant la station fixe, afin d'attendre que le transfert d'information soit effectué.

Connexion de plusieurs clients à un serveur

Dans notre troisième expérience, nous allons mesurer l'impact du nombre de clients qui se connectent au même serveur, sur le délai requis pour que les clients reçoivent un fichier provenant d'Internet. Pour mener à bien cette expérience, nous avons exécuté des clients l'un après l'autre, qui se sont connectés à un serveur de l'environnement de test, afin de mesurer le délai requis pour que le serveur les détecte et qu'ils reçoivent un fichier provenant d'Internet.

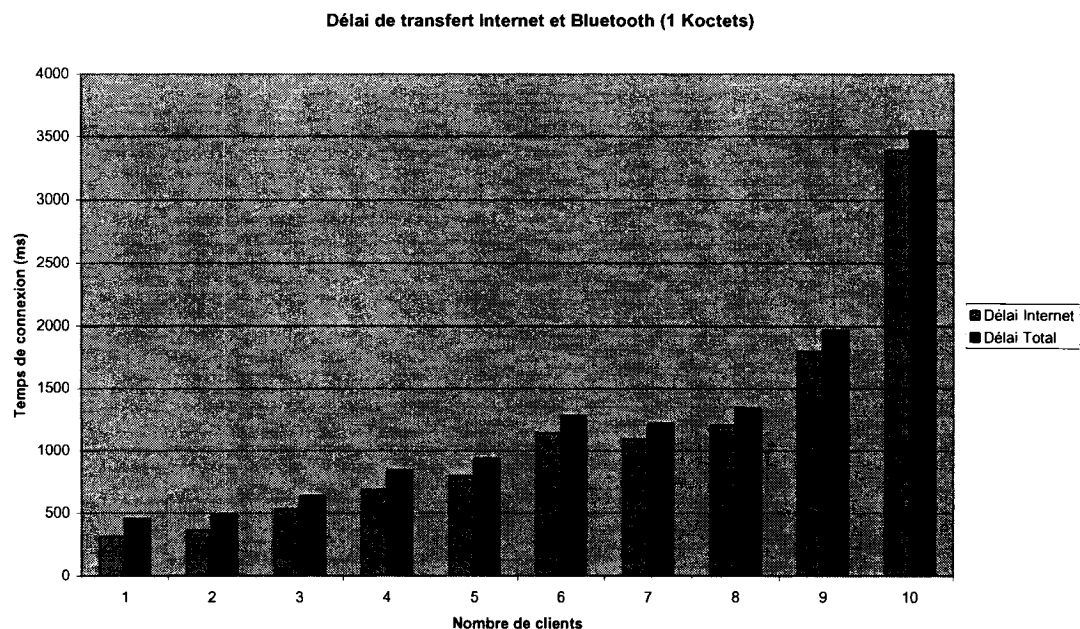


Figure 4.15 Délai de transfert de plusieurs clients à un serveur

Les mesures de notre expérience que nous avons commencée à prendre en compte après 5 sessions, sont présentées à la Figure 4.14. Nous voyons qu'en augmentant le nombre de clients, le délai de transfert Internet varie de 313 ms à 3390 ms

pour 10 clients connectés en même temps. Il varie aléatoirement de 110 à 150 ms pour la connexion Bluetooth. Ces résultats montrent que le délai de transfert pour les clients qui se connectent à un même serveur, dépend du nombre de clients déjà connectés. De plus, ces résultats nous montrent que c'est le délai induit par la connexion Internet, qui est le plus affecté par le nombre de clients déjà connectés.

Adaptabilité par rapport à la plate-forme d'exécution

Un des requis que nous avons identifié au Chapitre 3 est l'adaptabilité de l'interface graphique de notre application à son contexte d'exécution. Pour ce faire, voici un exemple d'utilisation où les usagers rentrant chez eux ont automatiquement sur leur téléphone mobile un résumé de leur prochain cours et les prochains chapitres de cours à lire. Ce même scénario est répété lors du cours où des informations adaptées peuvent être envoyées aux étudiants.

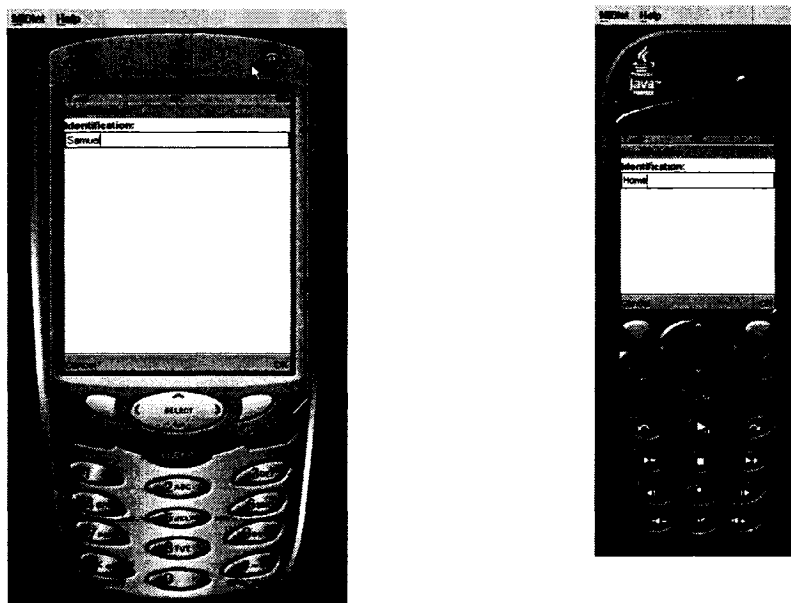


Figure 4.16 Identification des deux utilisateurs

Comme nous pouvons le voir à la Figure 4.16, les deux utilisateurs fournissent leur identificateur qui sera utilisé lors de la phase de connexion, à l'aide du clavier du téléphone. Ensuite, les deux utilisateurs choisissent leur mode de connexion qui leur permet soit d'attendre une connexion, soit d'être à la recherche d'un ou de plusieurs serveurs comme présenté à la Figure 4.17.

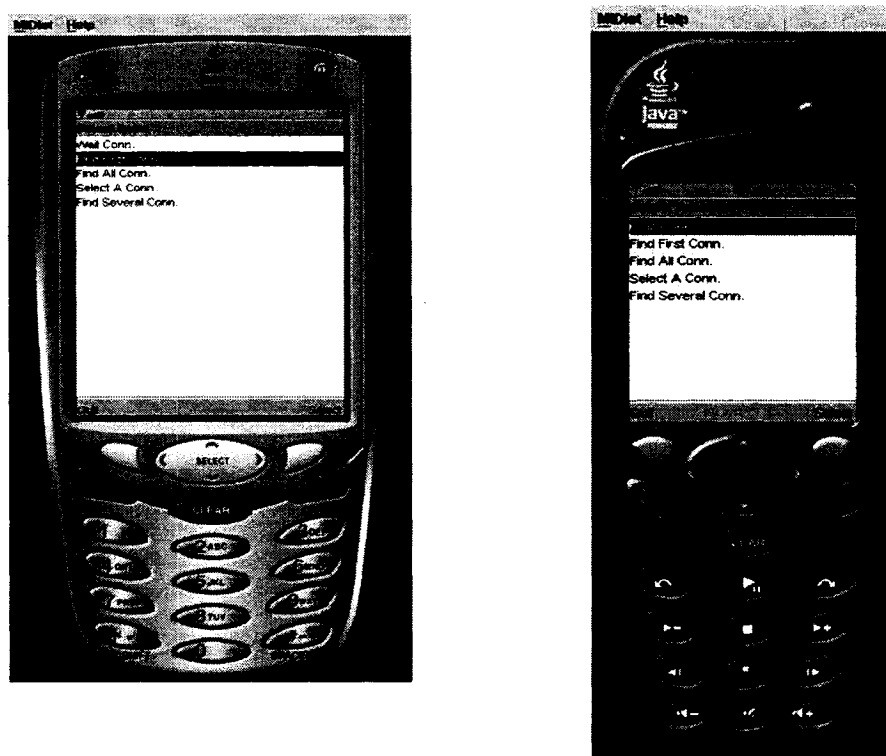


Figure 4.17 Choix du mode connexion des utilisateurs

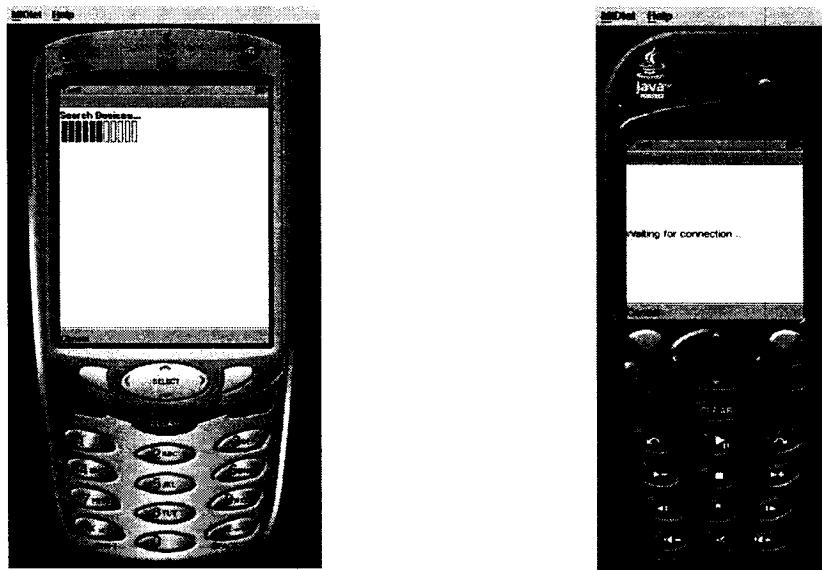


Figure 4.18 Recherche d'appareil mobile

À la Figure 4.18, l'utilisateur qui met son application en mode client est à la recherche d'un appareil mobile, et l'application en mode serveur attend un nouveau client.

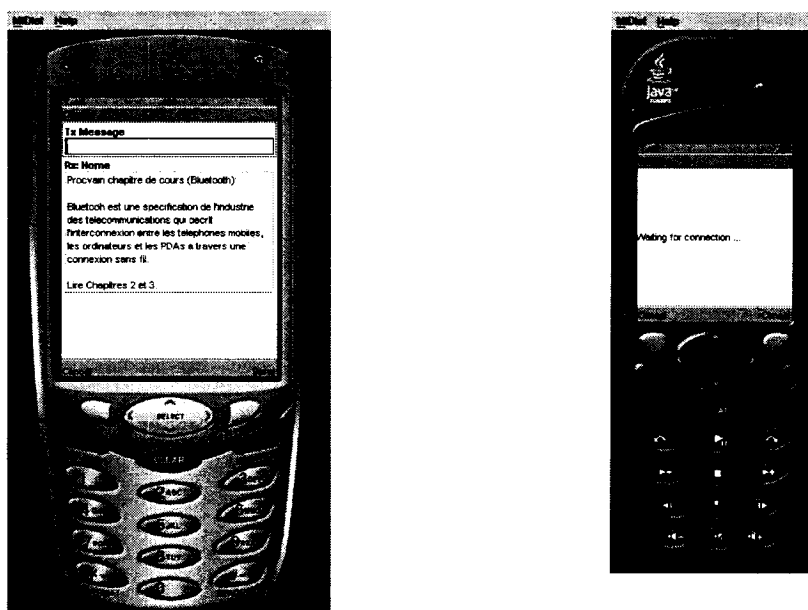


Figure 4.19 Détection et envoi de fichier

Enfin, lorsque le client se connecte au serveur branché à Internet à la Figure 4.19, ce dernier télécharge le fichier approprié et l'envoie au client.

Synthèse des résultats

Les différents scénari de test que nous avons implémentés, et les mesures que nous avons effectuées nous ont fourni des informations qui impactent la mise en œuvre de notre concept. En effet, nos résultats nous montrent que le délai de transfert global est beaucoup plus affecté par le débit de la connexion Internet que par la connexion Bluetooth. Nous pouvons déduire de ce constat que les serveurs qui requièrent une connexion Internet, doivent avoir une grande bande passante afin de ne pas constituer un goulot d'étranglement pour notre modèle. De plus, nous avons constaté que la variation de la taille des fichiers transférés a un plus gros impact sur la connexion Internet que sur la connexion Bluetooth.

En ce qui concerne un client qui se connecte à plusieurs serveurs, nous avons constaté que le nombre de serveurs est critique pour le temps de transfert requis. En effet, lorsque trop de serveurs sont présents dans la même zone géographique et qu'un client se connecte à tous les serveurs en même temps, le temps de transfert Bluetooth augmente de manière très significative pour les fichiers de petite taille. Cependant pour les fichiers de plus grande taille, cette augmentation du délai n'est pas significative.

4.4 Comparaison et évaluation des modèles

Nous allons dans cette partie de mémoire, confronter notre modèle décrit dans le chapitre 3, aux résultats que nous avons obtenus de l'implémentation de notre concept. Notre but est de montrer la possibilité de concevoir une plate-forme d'apprentissage mobile adaptable en fonction du contexte géographique d'apprentissage. Pour ce faire, nous avons identifié des requis que notre application devra respecter. Nous avons deux types de requis. Les requis d'apprentissage et les requis liés au développement de notre prototype d'application. Les requis d'apprentissage que nous avons identifiés sont les suivants :

1. le contenu éducatif est dynamique : les étudiants ont besoin de données mises à jour et ont besoin de communiquer avec des sources d'information variées ;
2. le contenu éducatif est opéré en temps réel : les étudiants peuvent avoir besoin de l'information de n'importe où et en tout temps ;
3. le contenu éducatif est collaboratif : étant donné que les étudiants peuvent apprendre les uns des autres, les applications d'apprentissage doivent pouvoir les connecter à des bases de données, à des professeurs, des experts ou d'autres étudiants ;
4. le contenu éducatif est personnel : chaque étudiant doit pouvoir sélectionner l'information dont il a besoin à un certain moment ;
5. le contenu éducatif est complet : les étudiants doivent avoir la possibilité de recueillir l'information dont ils ont besoin de plusieurs sources.

Pour répondre au premier requis, nous avons conçu notre modèle d'apprentissage de manière à ce que le contenu d'apprentissage soit mis à jour par les professeurs. De cette manière, nous nous assurons de la crédibilité et de la synchronisation des données transmises. Les professeurs mettent à jour l'information au niveau du serveur de données sur Internet. Cette information peut être modifiée en tout temps et dès qu'un nouvel utilisateur se trouvant dans un contexte géographique se connecte, l'information adéquate et mise à jour lui est automatiquement transmise.

En ce qui concerne le deuxième requis, nous l'avons aussi rempli. En effet, lorsque l'étudiant mobile est dans un lieu géographique, ce dernier reçoit la dernière information qui a été mise à jour au niveau du serveur Internet et qui est relative à l'endroit où il se trouve. De cette manière, les étudiants reçoivent la bonne information au moment opportun.

Nous avons répondu au troisième requis en créant un modèle permettant la communication entre étudiants ou entre étudiants et professeurs. En effet, les étudiants appartenant à un même cours peuvent s'envoyer de l'information ou communiquer avec

leur professeur afin de connaître ses heures de rendez-vous, ou avoir plus d'information sur un sujet spécifique.

Le quatrième requis est automatiquement rempli par notre modèle, étant donné que lorsqu'un utilisateur mobile se connecte à un serveur, la première opération effectuée est l'identification du serveur par le client et vice-versa. Ainsi le serveur adapte son fonctionnement et le contenu envoyé au client qui est connecté.

Enfin le dernier requis dépend de l'implémentation faite de notre modèle d'apprentissage. Étant donné que l'information qui est transmise et affichée provient d'un serveur Internet, nous pourrions adapter notre architecture pour que le serveur Internet puisse, en fonction de l'identification de l'application cliente (étudiant) et du serveur (point de connexion fixe), faire une recherche sur Internet afin de trouver l'information provenant de diverses sources, à transmettre.

Les requis d'implémentation que nous avons identifiés sont en fonction de la plate-forme d'exécution de notre modèle :

1. La taille des écrans : tel qu'illustré à la Figure 4.16, nous avons testé notre modèle sur plusieurs types de téléphones offrant des caractéristiques d'affichage différentes. Étant donné que la plupart des appareils mobiles offrent une taille d'écran et des possibilités d'interaction très limitées, nous avons conçu un modèle requérant un minimum d'interaction homme machine. En effet, à partir du moment où l'utilisateur démarre l'application, celle-ci pourrait fonctionner sans pour autant que l'utilisateur ait besoin de manipuler son appareil pour recevoir ou envoyer de l'information. Notre modèle permet aussi aux utilisateurs de définir une liste de serveurs auxquels leur appareil pourrait se connecter, ce qui permettrait de rajouter un niveau de sécurité à notre modèle.
2. La consommation d'énergie : nous avons conçu notre modèle afin qu'il puisse utiliser de manière optimale sa source d'énergie. La plate-forme *Midlet* de J2ME que nous avons utilisée, nous permet de placer notre implémentation dans plusieurs modes d'exécution qui réduisent leur utilisation des ressources énergétiques de l'appareil.

3. La puissance de calcul : en ce qui concerne l'utilisation optimale de la puissance de calcul des appareils, nous avons opté pour un modèle d'application où l'essentiel des calculs est fait au niveau du serveur Internet. Les deux appareils mobiles ne font que s'identifier, et c'est au serveur Internet de déterminer l'information adéquate à leur envoyer en fonction de l'identification du client.
4. Les capacités de communication : nous avons choisi d'utiliser un protocole sans fil et non payant pour l'essentiel des transferts de données. En effet, nous aurions pu utiliser d'autres moyens de communication qui passeraient par les opérateurs de téléphonie mais dont le coût pourrait ne pas être supportable par les étudiants. Nous avons donc choisi le protocole Bluetooth qui est doublement bénéfique, étant donné qu'il permet l'identification des autres appareils, ainsi que le transfert de données.
5. La navigation : ce requis est automatiquement rempli, étant donné que les utilisateurs n'ont presque pas à manipuler leur appareil mobile pour recevoir de l'information d'apprentissage. L'essentiel des interfaces usagers qui ont été implémentées concerne les pages où l'utilisateur fournit ses informations d'identification. De plus, l'utilisateur peut à tout moment fermer l'application, recevoir un appel téléphonique, pendant l'exécution de notre application ou changer ses paramètres. Notre implémentation permet aussi de passer d'un mode client à un mode serveur.

En somme, nous avons présenté dans ce chapitre l'implémentation de notre modèle d'application d'apprentissage mobile adaptable au contexte décrit dans le Chapitre 3. À la suite de quoi, nous avons réalisé une étude de performance de notre prototype d'application. À partir des résultats de cette évaluation, nous avons analysé et jugé de la possibilité d'implémenter notre architecture. Il ressort de cette analyse que notre modèle d'application d'apprentissage mobile adaptable est en mesure de répondre adéquatement aux requis que nous nous sommes fixés, même si des améliorations peuvent être envisagées.

CHAPITRE V

CONCLUSION ET DISCUSSION

Le problème de l'adaptabilité des applications d'apprentissage mobile a été abordé dans ce mémoire. Nous avons présenté puis évalué une solution basée sur le protocole de communication Bluetooth et la plate-forme d'exécution Java qui garantit la portabilité de notre modèle. Dans ce chapitre de conclusion, nous présentons dans un premier temps une synthèse des travaux et des résultats obtenus. Puis, nous abordons les limitations de notre solution, avant d'indiquer quelques pistes pour les améliorations et recherches futures.

5.1 Synthèse des travaux

Le problème que nous avons abordé est vaste et peut être résolu par différentes approches. Notre concept est plutôt basé sur la technologie J2ME. En effet, un de nos requis principaux est la portabilité de notre application et l'adaptabilité de son interface graphique. Nous avons utilisé le protocole Bluetooth pour la détection du contexte d'utilisation et pour la transmission des données entre utilisateurs mobiles ou entre un mobile et une station de base fixe.

Afin de pouvoir évaluer la performance de notre concept, nous l'avons implémenté dans l'environnement de test fourni avec NetBeans qui offre les mêmes fonctionnalités que les plates-formes mobiles. Nous avons développé un prototype d'application mobile configurable et pouvant fonctionner en mode client ou en mode serveur. Notre plate-forme de développement nous a permis d'exécuter plusieurs instances de la même application et de simuler la communication entre les instances. Ceci nous a permis de mesurer les délais de détection et de transfert de fichiers entre applications mobiles. Pour la majeure partie de nos mesures, nous avons utilisé un ou

plusieurs clients, un ou plusieurs serveurs, et un serveur Internet à partir duquel nous avons téléchargé les fichiers. Une analyse de performance, nous a permis de juger de la possibilité d'implémenter notre modèle. Après avoir identifié les indices de performance de notre application, nous avons réalisé nos mesures. L'analyse de ces mesures, nous a permis de constater que notre modèle d'application peut être utilisé dans un environnement d'exécution réel.

5.2 Limitations des travaux

L'adaptabilité des applications d'apprentissage est un problème qui présente plusieurs facettes. Dans le cadre de ce travail, nous nous sommes concentrés sur l'aspect de la détection du contexte d'utilisation et sur le transfert de l'information entre appareils mobiles. Cependant, pour des applications de notre modèle qui seraient de plus grande envergure, il pourrait arriver qu'il y ait tellement de contextes d'utilisation possibles, que l'identification du contexte soit très complexe. Il s'agit donc pour faire face à ce problème, d'implémenter des mécanismes qui vont permettre d'identifier le contexte d'utilisation au niveau d'un serveur Internet qui serait plus performant qu'une plate-forme mobile. Une piste serait donc d'avoir un serveur Internet qui serait doté d'une plate-forme d'Intelligence Artificielle pouvant déterminer l'information adéquate dont auraient besoin les utilisateurs mobiles.

La seconde limitation de l'étude réside dans le fait que nous avons utilisé un environnement d'exécution virtuel pour faire nos mesures. En effet, il peut arriver que les appareils mobiles aient une plate-forme d'exécution Java, mais n'aient pas assez de mémoire ou de puissance de calcul, pour exécuter notre application. Afin de combler cette lacune, il s'agirait de choisir différents modèles d'appareils mobiles sur lesquels nous installerons notre application et ainsi effectuer les mesures de performance dans un environnement réel d'exécution.

Notre troisième limitation vient du fait que notre modèle ne peut être appliqué que sur un appareil mobile disposant d'une plate-forme Java ainsi que de la technologie Bluetooth. Nous avons choisi d'utiliser la plate-forme Java pour la portabilité de ses

applications, et Bluetooth nous permet de localiser les utilisateurs mobiles et de leur transférer des données gratuitement. Cependant avec l'avènement de nouvelles technologies telles que le *Mobile Positioning System (MPS)* développé par Ericsson (ERICSSON), d'autres moyens de localisation sont actuellement implémentés par les concepteurs d'appareils mobiles.

En ce qui concerne la quatrième limitation, elle concerne essentiellement le fait que notre étude ne porte pas sur le contenu d'apprentissage mais sur la technique permettant le transfert de l'information aux étudiants ou aux professeurs. Nous pourrions envisager d'approfondir notre recherche en menant un sondage auprès d'étudiants et de professeurs afin de connaître leurs besoins en termes de contenu d'apprentissage mobile. Et afin de confirmer les données recueillis, choisir un groupe d'étudiants et de professeurs qui pourraient utiliser notre plate-forme d'apprentissage afin de tester les différents scénarii d'apprentissage.

5.3 Indications de recherches futures

Les résultats de mesure que nous avons obtenus nous montrent que notre modèle d'application d'apprentissage mobile adaptable au contexte d'utilisation peut être implémenté dans des téléphones cellulaires, afin d'aider les étudiants à mieux gérer les ressources qui sont à leur disposition. Nous pourrions penser à une université qui mettrait en place des points de connexion Bluetooth qui transféreraient aux étudiants, des informations sur mesure, telles que leur emploi du temps, le résumé de leurs différents cours, ainsi que les événements majeurs rythmant la vie étudiante.

De manière plus générale, le protocole Bluetooth pourrait être utilisé comme lien de connexion entre le contenu d'Internet et les appareils mobiles. Les universités pourraient recueillir le profil de tous les étudiants disposant d'un téléphone avec la technologie Bluetooth afin de leur fournir de l'information adaptée à leur localisation et à leur profil.

BIBLIOGRAPHIE

CHEN, G., KOTZ, D., “A Survey of Context-Aware Mobile Computing Research”, Department of computer science, Dartmouth College, TR2000-381, November 2000.

EISENSTEIN, J., VANDERDONCKT, J., PUERTA, A., “Adapting to Mobile Contexts with User-Interface Modeling”, Third IEEE Workshop on Mobile Computing Systems and Applications, pp. 83-92, December 2000.

EISENSTEIN, J., VANDERDONCKT, J., PUERTA, A., “Applying Model-Based Techniques to the Development of UIs for Mobile Computers”, International Conference on Intelligent User Interfaces, Santa Fe, New Mexico, United States, pp. 69-76, 2001.

GAEDKE, M., BEIGL, M., GELLERSEN, H., SEGOR, C., “Web Content Delivery to Heterogeneous Mobile Platforms”, {ER} Workshops, pp. 205-217, 1998.

GELLERSEN, H., SCHMIDT, A., BEIGL, M., “Multi-Sensor Context-Awareness in Mobile Devices and Smart Artefacts”, ACM journal Mobile Networks and Applications (MONET), vol. 7, n° 5, October 2002.

GEORGIEV, T., GEORGIEVA, E., SMRIKAROV, A., “M-Learning - a New Stage of E-Learning”, International Conference on Computer Systems and Technologies (CompSysTech'2004), Session 4, 2004.

HAARTSEN, J., NAGHSHINEH, M., INOUE, J., JOERESSEN, O., WARREN, A., “Bluetooth: Vision, Goals, and Architecture”, Mobile Computing and Communications Review, vol. 2, n° 4, pp. 38-45, 1998.

HIX, D., “Generations of User Interface Management Systems”, IEEE Communications Magazine, vol.7, n° 5, pp. 77-87, 1990.

JACOB, R., “Open Syntax: Improving Access for All Users”, Workshop on Universal Accessibility of Ubiquitous Computing, Portugal, pp. 84-89, ACM Press, 2001.

KLEIN, M., KÖNIG-RIES, B., “An Ontology-Based Document-Space as an Adaptable User Interface for Mobile Information Systems”, Mobile Datenbanken und Informationssysteme, 6th ed, Germany, University of Karlsruhe, pp. 10-14, 2002.

LUYTEN, K., VAN LAERHOVEN, T., CONINX, K., VAN REETH, F., “Specifying User Interfaces for Runtime Modal Independent Migration”, Computer-Aided Design of User Interfaces, CADUI’2002, May 15-17, Université de Valenciennes, France, 2002a.

LUYTEN, K., VANDERVELPEN, C., CONINX, K., “Migratable User Interface in Component-Based Development”, Interactive Systems: Design, Specification, and Verification, Vol. 2545 of Lecture Notes in Computer Science, pp. 62-76, 2002b.

MITROVIĆ, N., MENA, E., “Adaptive User Interface for Mobile Devices”, 9th International Workshop DSV-IS 2002, Portugal, Springer Verlag Lecture Notes in Computer Science LNCS, ISBN 3-540-00266-9, pp. 47-61, 2002.

PRIBEANU, C., LIMBOURG, Q., VANDERDONCKT, J., “Task Modelling for Context-Sensitive User Interfaces”, Lecture notes in computer science, In Proceedings of the Eight Workshop of Design Specification and Verification of Interactive Systems, pp. 60-76, June 2001.

SCHMIDT, A., “Implicit Human Computer Interaction Through Context”, Personal Technologies, vol. 4, n° 2, June 2000.

VAN, LAERHOVAN, K., "Technology for Enabling Awareness", In Site TEA, <http://www.teco.edu/tea/>, 2000.

WANG, Y., "Context Awareness and Adaptation in Mobile Learning", IEEE Computer Society on WMTE, pp. 154-158, 2004

SITES WEB

ECLIPSE : www.eclipse.org

EJB : <http://java.sun.com/products/javabeans/index.jsp>

ERICSSON : http://www.ericsson.com/mobilityworld/sub/open/technologies/mobile_positioning/index.html

JAVA : <http://java.sun.com>

J2EE : <http://java.sun.com/j2ee/index.jsp>

J2ME : <http://java.sun.com/j2me>

J2SE : <http://java.sun.com/j2se>

JSP : <http://java.sun.com/products/jsp>

Jsr118 : www.jcp.org/aboutJava/communityprocess/final/jsr118

Midp : <http://java.sun.com/products/midp>

NETBEANS : <http://www.netbeans.org>

NOKIA : www.nokia.ca/french/products/7610/7610.asp

O'REILLY : www.xml.com

SERVLET : <http://java.sun.com/developer/onlineTraining/Servlets/Fundamentals>

SOURCEFORGE : <http://sourceforge.net/projects/jxul/>

SYMBIAN : <http://www.symbian.com>

Windows : <http://msdn.microsoft.com/library/fre/default.asp?url=/library/fre/vbcon/html/vbconserverbasedtimers.asp>

WINMOB : <http://www.microsoft.com/windowsmobile/default.mspx>