Scientific Research

Mobile Applications Provisioning Using Bluetooth Wireless Technology

Thomas Djotio Ndie[1], Claude Tangha[2], Tam Sangbong[3], Armstrong Formenah Kufor[3]

[1]LIRIMA, Management and Security of Network and System Services, University of Yaounde I, Yaounde, Cameroon; [2]Department of Computer Science, National Advanced School of Engineering, University of Yaounde I, Yaounde, Cameroon; [3]Soft-Tech International Inc, Douala, Cameroon.
Email: {tdjotio, ctangha}@gmail.com, tam@soft-techint.com, kufamstrong@yahoo.com

ABSTRACT

In this paper, we propose a cheap means for propagating mobile application using Bluetooth, a convenient short range wireless technology. Today, one of the main problems about mobile technology is about implementing automatically the software in the multiple and various type of phones devices, irrespective of the phone model (Nokia, Samsung, iPhone, etc.). There are many ways of doing OTA like by SMS. This approach allows providing an http download link by SMS but, we are limited by the maximum size one can have to build an application and send it trough this means. This approach is also expensive because data sources are expensive. Within the context of mobile social networking and proximity marketing, we come up with an original way of provisioning mobile applications using Bluetooth. We use anybody who is already a subscriber of an application to become an ambassador of it. The application is self-replicating and distributing in itself and sending back results. It ethically acts like a virus or a disease. A prototype is built to validate the proposed methodology.

1. Introduction

We propose to Software Designing Companies in the mobile application development field a short range provisioning method of mobile application for advertisement, proximity marketing and other motives for propagating information independent of any exiting mobile operator and model. Indeed, cost of information propagation, advertisements, sensitization and other purposes of propagating information through information technology (IT) has become a cause for concern for most organizations and individual. Nowadays, these organizations and individuals mainly target other individuals (the clients) of advertisement, mostly those in possession of at least a mobile phone. Most commercial operators also make use of the mobile operators' services in order to advertise the value-added services they offer to their customers. Likewise, application developers deploy their applications on servers, allowing mobile customers to browse the web in other to download them. Some of these information propagator media are somehow very costly, therefore affecting most of these organizations. Minimizing the cost of propagating information by most of these IT technologies

thus becomes a major cause for concern by most of these organizations [1-3]. In both cases, the operators or the mobile client have to go through an expensive middle man (Internet or mobile operators) irrespective of the distance between transceivers. And also, most mobile phones do not integrate discovery applications like web browsers, GPRS etc. that allows for easy browsing and downloading of application.

The provisioning of mobile applications in the context of today is very expensive as users have to purchase, and then download them from the Internet which cost a lot of time and hence money. Most mobile applications also have a limited market due to their limitations to certain platforms. The challenge is to make a difference in the case of short range advertisement, precisely to bridge the "expensive middle man" in this context [1]. For the purpose of proximity marketing for example, commercial operators will want to call to the attention of his customers by pushing messages of any form onto the phones of such clients who we suppose are at a certain distance (for example most 100 meters) from him, this without depending on the services of any third parties (like mobile

operators) [3,4]. But how does this commercial operator detect the presence of his targets, and push these messages to them? The most common and current technique used today is messages-based advertisement method through SMS. Advertisement methods that are based on the short range wireless technologies through applications (that will advertise the services offered by the provider) still to come and to develop [5-8]. Our original approach contributes in this second area and is based on the Bluetooth wireless technology [9-11]. It is indeed an opened wireless standard that is increasingly being integrated in mobile devices (and almost every adult today is in possession of a mobile phone). Other important reasons motivated us for adopting this wireless standard for provisioning of mobile applications. Firstly it uses an unlicensed frequency hence independent of any mobile operators and thus cheap. Secondly, many manufacturers are integrating this technology into their products; ranging from mobile phones, TVs, cars just to name a few [7]. As such, using such a wireless technology will simply boast the size of the targeted market. Thirdly and most importantly, this short range provisioning method is used in most case to advertise information to a group of individuals in a given area, with an intended purpose, considered here as a mobile social network [7]. This is one of the most important branch of proximity marketing and its extensions: broadcasting formation on school campus, museum, conference etc.

Given the challenge, these reasons and objectives, we propose a self-replicating and distributing methodology for mobile application provisioning. The application is ethically continued to spread or auto-replicate by itself like a virus or a disease, by the help of the technology which by itself will do self propagation. The principle of our approach is as follow. Within an environment where there are many Bluetooth compatible phones, anybody who is already a subscriber of the application becomes an ambassador of it. Once a new phone or an Up-In person is detected thanks to the Bluetooth signal, the subscriber is going to use a profile which is a Java 60kb MIDlet, to ask the Up-In person if he wants the subscriber to install the application on him. For the ethical purpose, the acknowledgement process cannot be bypassed. It is like asking someone if he would like to participle to a game in the context of social networking. If he says "YES", he will now download and install a MIDlet profile which will now read all the characteristics of his phone and then communicate that with the host phone. The phone now will know who that phone is, what kind of phone it is, so it now knows what version of the application might be needed to move. The application will now transport itself from this phone to the next phone but following the device type that the person has. To address the problem, we start in the Section 2 by studying the existing short range

wireless technologies by focusing on Bluetooth. The Section 3 presents the solution proposed. The Section 4 presents a prototype built to validate the approach. The paper ends up with a conclusion and future works but before, we discuss our solution in Section 5.

2. State of the Art

In this section, we set out to study the existing wireless technologies that are integrated in mobile phones. We start by presenting the choice of the technology to adopt. We then describe the existing solutions for provisioning. We end up by the presentation of the existing solutions for advertisement.

2.1. Choice of the Technology to Adopt

The most common wireless technologies integrated into phones by most mobile phone vendors today are Wi-Fi [1,2], Infrared, Zegbee [8,12,13] and Bluetooth [7,14,15]. A summary of them is presented on the table 1 below. The criteria used for selecting the technology to be used focused on the autonomy of the mobile phones, and the fact that users of these devices are frequently mobile. As such, interest was focused on properties (1) and (6) (see **Table 1**): communication media and power consumption.

With these properties as focused, we found out that WI-FI was unsuitable for our methodology since it is high in energy consumption (inducing low autonomy in phones using this technology) [1,2]. On the other hand

Table 1. Comparison between short range wireless technologies.

Technology→ Properties	WI-FI	Infrared	Bluetooth	Zegbee
Frequency band (1)	2.4 GHz	-	2.4 GHz	2.4 GHz, 868 MHz, 915 MHz
Communication media (2)	Radio waves	Light rays	Radio waves	Radio waves
Throw output (Raw data rate) (3)	11-54 MB/s	1 MB/s	1 MB/s	250 KB/s
Data type (4)	Digital		Digital, audio	Digital, key-value pairs
Inter-node range (5)	100 m	0-10 m	10-100 m	10-100 m
Power consumption (6)	Few hours on one battery (typically 0-6 hrs)	Week/Months on one battery	Few days on one battery	Years on one battery
Current Mobile market penetration (7)	Low	Medium	High	Very low (emerging)

Infrared which uses light rays for propagation has this in convenience that transceivers have to be in "**a line of site**" for communication to be possible (which is practically impossible since user of the mobile phones are in movement) [7]. Having discarded Wi-Fi and infrared, the leading competitors are Bluetooth and Zegbee [8]. Though Zegbee is relatively economical in terms of energy consumption, it is still an emerging technology and has not yet gained ground in the mobile market [12,13]. Bluetooth is therefore adopted reasons being that 90% if not more of the phones in the world market nowadays integrate this technology into their system [1,15].

2.2. Existing Solutions for Provisioning

To use Bluetooth to meet up our objectives, we had to study the current state of art for provisioning of mobile applications to mobile clients. The study then serves as a guide to model our own system for advertisement.

2.2.1. Classical OTA (over the Air) Provisioning

Here the mobile client initiates the whole process by discovering the application over a wireless network using a suitable discovery application like a WAP; web browser etc. The user uses the HTTP protocol or GPRS to discover applications over a wireless network and then downloads them (both application descriptor file and application archive file) [3,6]. As inconveniences, 1) the user of the mobile phone has to go through the painful process of discovering applications. 2) Most mobile phones in third world countries like Cameroon do not yet integrate these expensive discovery applications like WAP, web browser, GPRS etc.. 3) Downloading the application is expensive since the user uses an expensive network to get the application and secondly, if the application is commercial, he has to pay for it (pay for the network and then, pay again for the application).

2.2.2. WAP Push Provisioning

Here, the user of the mobile phone uses a convenient tool like a PC which has a web browser to discover the application very the network (Internet etc), enters require information about the mobile phone [7]. The provisioning server then takes care of pushing the application onto the mobile phone using the HTTP protocol. For pushing to be possible the mobile phone in question has to integrate the same expensive discovery application into their systems. This means WAP push provision shares some of the inconveniences of the Classical OTA (over the air provisioning).

2.2.3. Local Provisioning

In cases where connections to an expensive network or expensive discovery applications are undesired, other

provisioning methods were designed amongst which we have: **1) Device connectivity software** where most mobile phones come along with software that allows for sharing of data between PCs and mobile phones. This software is therefore used to transfer downloaded or developed application onto mobile phones [16]. **2) Bluetooth Infrared connectivity** where some of the mobile phones that are Bluetooth or infrared enabled also use a similar process to share data between PCs and mobile phones. A good example of software is Motorola phone tools [16,17]. The main drawback of the local provisioning methods is that they are all usually vendor dependent. The user of software connectivity for a Nokia phone for instance will not be compatible for a Samsung phone.

2.3. Existing Solutions for Advertisements: Case Study of Cameroon

The propagation of information: use of images, sounds, slogans, or any form of digital signal to communicate message that will spark targets interest to the desired goal of the initiator has gained ground in the market-based economy and in the daily life of every citizen. This propagation allows commercial cooperators to promote products, politician to connect to voters, states to send information to its citizens, individuals to communicate between themselves etc.

In the particular case of Cameroon, to communicate with other mobile phones, individuals, co-operations must be used the services of one of the following GSM/CDMA networks: MTN, ORANGE and CAMTEL. The average cost of calls per minute stands today at FCFA 100 local currency (about $0.2) likewise for calls irrespective of the distance separating the transmitter and the receiver of the information in transit. Up growing local commercial operators like EXPRESS EXCHANGE, CREDIT COMMUNAUTAIRE D'AFRIQUE, EXPRESS MONEY GRAMME, just to name a few , make use of the services of these mobile operators to add quality to the services offered[1] to their clients who are usually in possession of mobile phones. But the cost of propagation of this information has a negative impact on the objectives of the co-operation which is profit maximization. In what follows, we will be presenting a means to eliminate the dependency of the services of these mobile operators in the case of short range advertisements. The difference being that we use applications (containing Icons, animations etc.) instead of messages.

3. Mobile Applications Provisioning Using Bluetooth Wireless Technology

Most commercial operators who need to target clients that are in the specific locations for a specific reason are facing some challenges: reducing the cost of provisioning

---

[1]They basically offer the money transfer service.

numeric data like mobile applications using information technology for proximity marketing, local advertisements and maximizing means of targeting customers with low or no financial expenses. In this section, we start by describing the problem. We then follow by presenting the targeted public. After describing the architecture of our solution, we talk about the system design and analysis, starting with the requirements (technical and functional) then identifying the use cases of the system and ending with the UML diagrams describing the system.

3.1. Problem Description

We build an original concept on something that exists: the Bluetooth wireless technology. We are looking for a suitable way of automatically and ethically distributing a mobile application within a mobile social network. In this group of people and especially in this context, anyone must have a Bluetooth enabled phone and it needs to be on. Within such environment, a phone is constantly propagating itself to all others. For example, in the context of m-learning, an application can propagate homework to mobile students.

The phone on which the application is already installed is called the subscriber (**A**). It sends/broadcasts a welcome/request message to all Bluetooth enabled devices, when it is in range which someone (**B**) who has its Bluetooth on. It can look like this: "I am this person A and I would like to send you this application, will you accept? Yes/No" This initial message is based on the Bluetooth standard. The other phone (**B**) does not yet have anything from the subscriber (**A**). (**B**) just needs to react/answer back by processing the standard two-sides binding handshake. If the answer is "Yes", (**A**) is not going to send the application but instead a 60 kb Java MIDlet generic profile [6,17]. That MIDlet profile is now going to read the IMIE number of the phone (**B**), identify it, also check the device manufacturer and then send these information back to subscriber (**A**), so it can now understand which phone (**B**) is. When (**A**) is getting into that information, it will now go into its application data store, a sort of assembly where all versions of application are keeping, for identifying the phone type of (**B**), the manufacturer, the version of the application compatible with (**B**), takes it back and installs it on (**B**). When he is installing it in the recipient phone (**B**) after it accepted to participate, it takes the entire data store and installs it too in it, so it is ready for the next operation. Indeed, by answering 'Yes', the next phone (**B**) accepted to be the ambassador of the application. It accepted to provision the application on other phones and to become a member of a mobile social network: case of the proximity marketing for example. We now have a self-replicating application.

To better understand the problem to address, we are going to illustrate it with a disease case-based example. In this case indeed, when a new person (the targeted device (**B**)) go in contact with a sick man (here the subscriber (**A**)), the disease (the mobile application) might be developed differently. The installed copy of a microbe contacted is according to the immune system (phone type) of the new person.

3.2. Targeted Public

The solution to propose will be used for the diffusion of information within the proximity of an advertising source (campus, supermarket, conference, museum etc.), a mobile social network group.

3.2.1. Users

The users of this solution or software are individuals, organizations, corporations, institutions that advertise their services at short ranges. A good example could be considered on a school campus where advertising emitters are hosted in each department to advertise information within the proximity of the department, a hundred meters around the building.

3.2.2. Clients

The source has as aim to attract the clients towards the services they are offering. A super market can therefore use this means of advertisement to attract client outside but within 100 m from the supermarket to advertise their product. This will then attract clients to the super market.

3.3. Architecture of the System

The approach for provisioning and advertisement proposed here has these particularities:

- Bluetooth is used in the place of the wireless network.
- The discovery functionality of the client is send to the server side, thus the client no longer needs to have discovery applications installed on his system. The only constrain being that the client mobile phone has to be Bluetooth enabled.
- The system we designed is basically client/server architecture:
- **Client**: this is the user of the device on which the application will be installed, He uses his Bluetooth enabled device to search for Bluetooth enabled devices within a radius of, 10meters (if host is a class one device), 20 meters (class two) or in the best of cases 100 meters (class three).
- **Server**: amongst devices discovered, the devices selected for pushing will be the servers of our architecture. This will be essentially mobile phones that support the OBEX object push services (FTP and OPP).

To achieve the functional objectives described above

and illustrated **Figure 1**, we make use of some of the protocols found in the Bluetooth protocol stack, namely **Service Discovery Protocol (SDP)** and **Object exchange protocol (OBEX)** [7].

- **Service Discovery Protocol (SDP)**: this is a protocol specific to the Bluetooth technology. It is specifically a peer to peer protocol. It is used for the discovery of devices and services found on the devices. The Bluetooth emitter will therefore use this protocol for the discovery of devices within its proximity, and then go further to search for push services found on the devices. The push services to be found are presented in the **Table 2**.
- **Object exchange protocol (OBEX)**: this protocol is used for the exchange of objects between Bluetooth enabled devices. It will implement the pushing functionality of the two profiles of **Table 2**.

Here the provisioning server installed in the advertising emitter found at the advertising source performs Bluetooth inquiries to discover Bluetooth enabled devices round its vicinity. Once found, the devices are filtered. Filtering here involved discarding those that are not in the category of mobile phones as depicted on **Figure 2**. After filtering, the emitter then searches for the connection URL found on the remote devices service discovery database (SDDB). The URL is returned in the following form [7,9-11]:
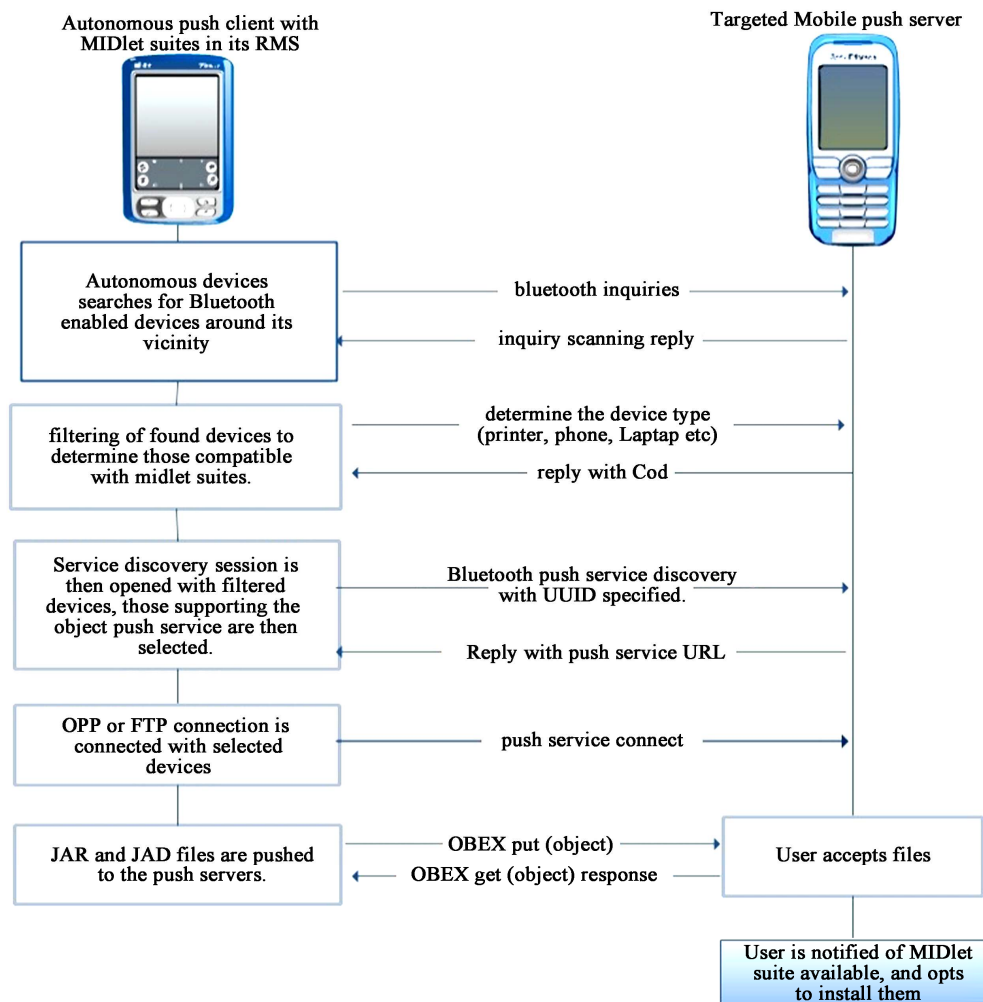
Profile//: *Bluetooth address*: *port*; *connection setting*
Example: ***Goep//*:1234968945:2; *master = true*; *authentication = true*; *encryption = true.***

Thus, to connect to the push services provided by the

Table 2. Push services to use.

Profile	Service Name	UUID (decimal)	UUID (hex)
Object Push	OBEX Object Push	4357	0×1105
Object Push	OBEX File Transfer	4358	0×1106

Figure 1. Proposed provisioning and advertisement.

Bluetooth Enabled devices (discoverable servers) on An Inquiry scan mode
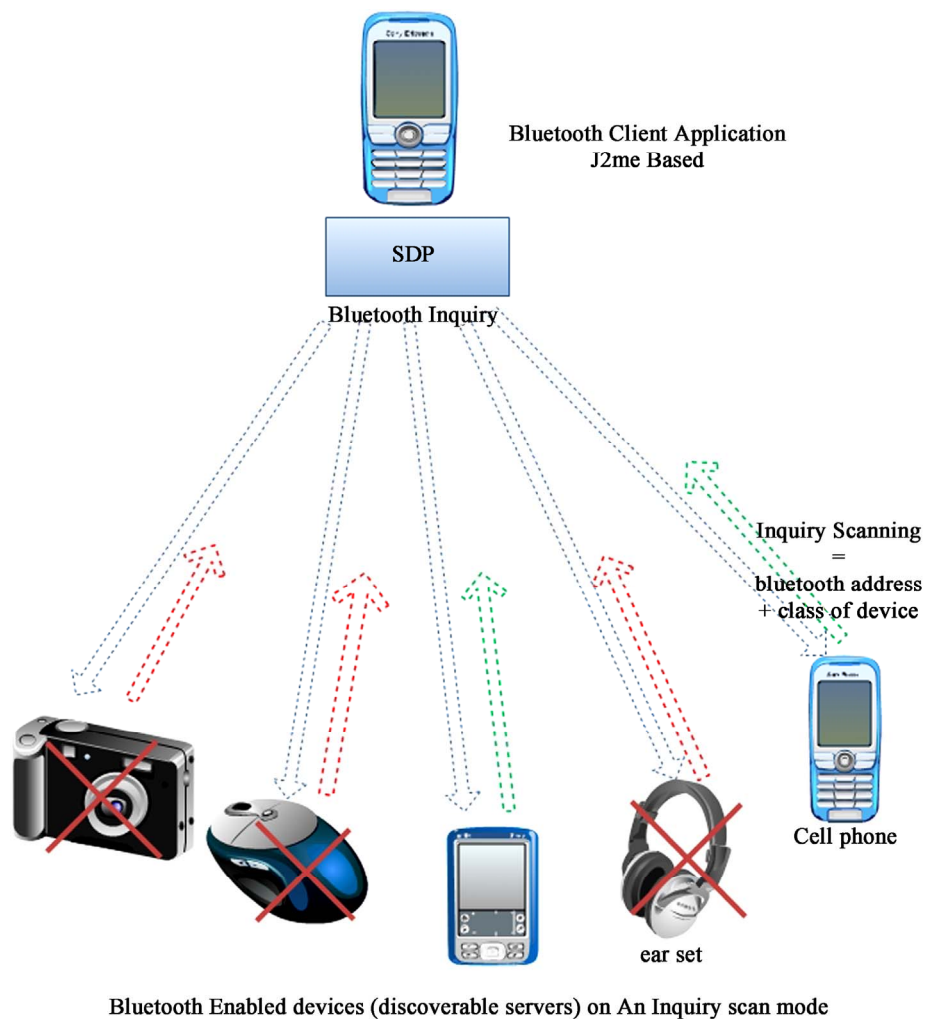
Figure 2. Discovery and filtering.

URL above, the provisioning servers has to make use of the generic object exchange protocol (GOEP) to connect to the Bluetooth address (1234968745) with an emulated serial port (2) indicated. And for connection to be possible it has to connect as a master device and accept authentication and encryption of data during transit. The process of push service search and provisioning is depicted on **Figure 3**.

1) The Bluetooth emitter uses the service discovery protocol to search for push services (having UUID 0×1106 and 0×1105), this discovery process is done without the awareness of the link security manager, thus the remote devices is unaware of the discovery process.

2) The UUID is queried in the SDDB of the remote device (B), and once found, the URL is returned to the autonomous device (A).

3) The Bluetooth emitter now uses this to connect and push application's JAR and JAD file to the inbox of the remote device.

3.4. System Analysis

The description above helps us to come out with the following UML analysis first starting with the requirement analysis and ending with the corresponding UML diagrams:

3.4.1. Functional Requirements
Functional requirements present the needs to be addressed by the software under development.

- **Device discovery and filtering**: this involves discovering remote devices within the vicinity of the autonomous device, and determines what type of devices they are (cell phone, smart phone, Laptop, printer etc), this in view of predicting if the application to be deploying will be compatible with the remote device. In the current context, the application is to be deployed on mobile phones having a human-machine interface, thus eliminating devices that are not of this category (see **Figure 2** above).

- **Discovering push services**: the autonomous device will have to determine those devices offering the OBEX push services (present in the OOP and FTP), in order to determine those to which it is going to push MIDlet suites (mobile application).
- **Profiling discovered devices**: this involves determining the model of the device discovered: its vendor and version. This in view of determining the version of the application to be deployed on the device.
- **Pushing MIDlet suites:** This is the final and most important function the application has to be designed. This is the streaming of the application and transporting it over the Bluetooth radio waves to remote devices, which will then install them.

3.4.2. Requirement Analysis

To meet up with these functional requirements we divide our system into the following modules:

- Discovery Module: performs the inquiry process to determine Bluetooth enabled devices around the vicinity of the host device (autonomous device).
- Filtering Module: is responsible for eliminating devices which are out of the context of the application. Devices like desktops, Laptops, printers will be eliminated.
- Push service discovery module: is responsible for searching in the SDDB of the remote device (discovery server) for push services and connecting to them.
- Profiling Module: this module (totally independent of the main application) will be pushed by the main application onto a remote device after a push service has been discovered by the autonomous device. Once installed on the remote device, this module (application) will report back to the main application, informing it of what model and versions the remote device is.
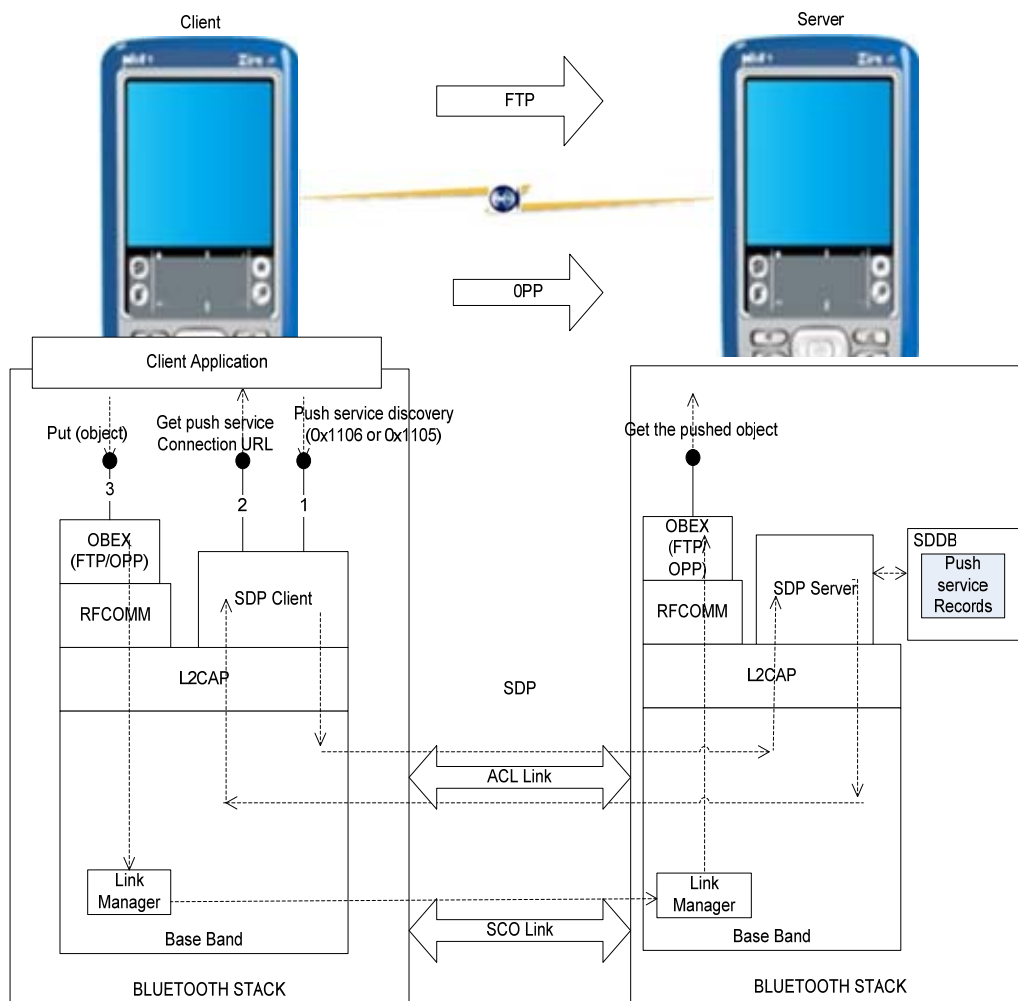

Figure 3. Push service discovery and provisioning.

3.5. System Design

From the requirement description seen above, we identify the actors of the system and different UML diagrams and the system architecture.

3.5.1. Actors and Use Cases
Here we come out with the actors and different use cases diagrams from the requirement analysis.

a) *Application's initialization of*
- **Actor**: User of Autonomous device
- **Description**: he intervenes during the launching of the application and the entering of PIN codes when need arises or switching the application to a manual mode to observe all the steps involved in the propagation of the MIDlet suites.

b) *Application use case*
- **Actor**: Bluetooth Application
- **Description**: the application itself will be looked upon as an actor, but this time around acting on an-

other system which is the Bluetooth protocols stack, upon which it will perform actions specific to Bluetooth applications, they include, device discovery, service discovery, filtering of devices, and pushing of objects.

c) *Profiling use case*
- **Actor**: Profiling module;
- **Description**: this module is independent of the main application, and will be deployed to the remote device for profiling. Once installed on the remote device, the will query for the runtime platform and versions of the remote device then report back to the main application.

3.5.2. Class Diagram
A static view of our system is shown **Figure 4**. The following classes are derived from the functional requirements:
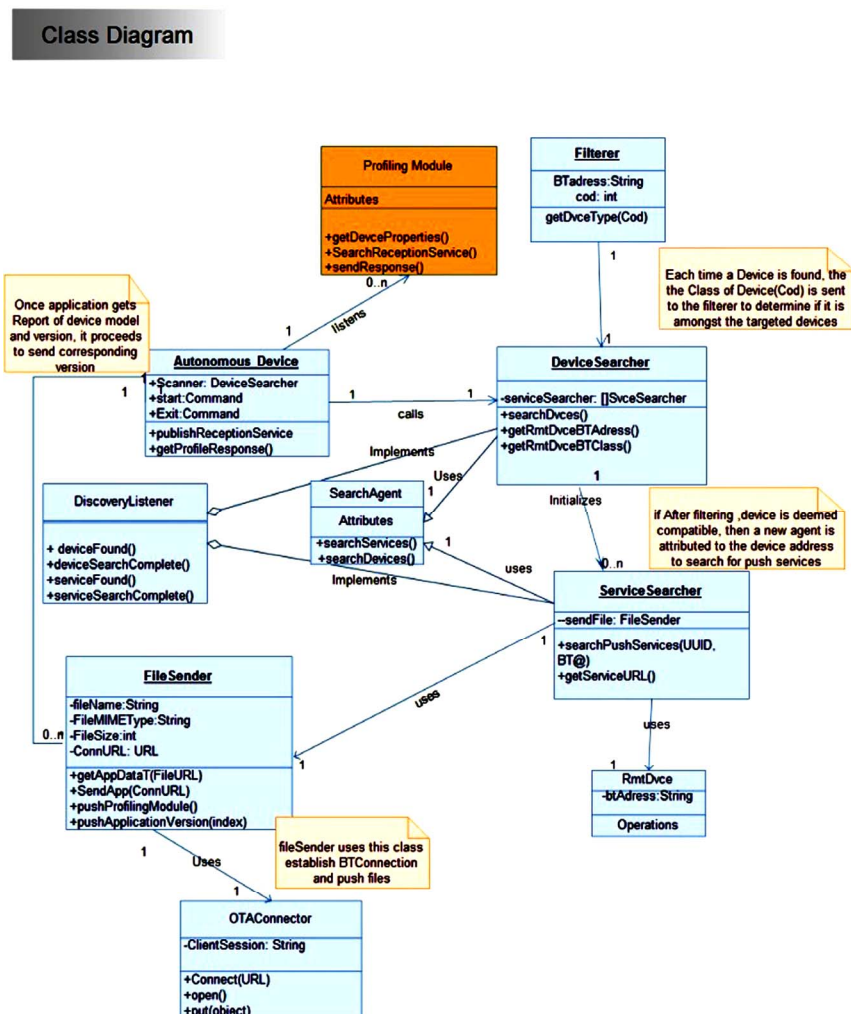- **Autonomous_Device**: this class represents an inter-

Figure 4. Class diagram (Static view).

face to all the functionalities of the application to be designed.

- **Device_Searcher**: this class is responsible for performing the Bluetooth inquiry process to discover remote devices in the vicinity of the Autonomous device.
- **Filterer**: this class will determine the type of device and if it is amongst the devices type targeted.
- **Service_searcher**: this class is responsible for searching push services (OPP and FTP), and obtaining the universal resource location (URL) for connection to these services.
- **File_Sender**: this class obtains the URL for connection, establishes a Bluetooth connection to the remote device, after connection it sends MIDlet suites

to the remote device.

- **Profiling_module**: this module could be looked upon as an independent application which will be pushed by the main application onto a remote device for profiling. Once it profiles the remote devices, it sends feedbacks to the autonomous device. The autonomous device then decides which version of the application to send to the remote device.

3.5.3. Activity Diagrams

We depict by these two activity diagrams two important aspects of the automation of the functionalities of the system. The **Figure 5** illustrates the **device and service discovery** activity and the **profiling and provisioning** concern is shown in **Figure 6**.
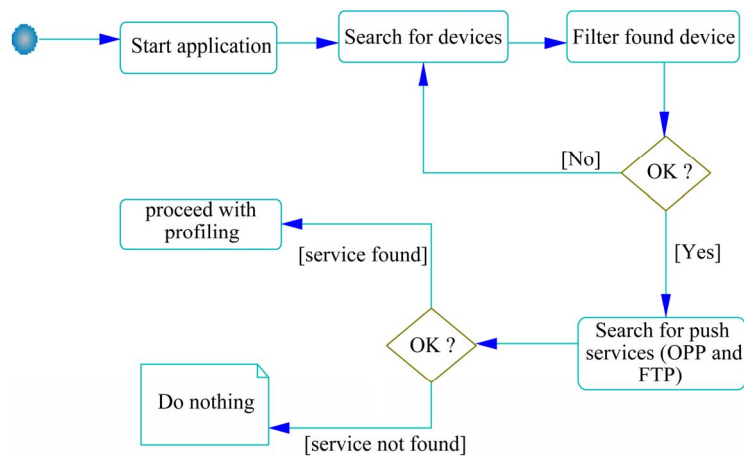

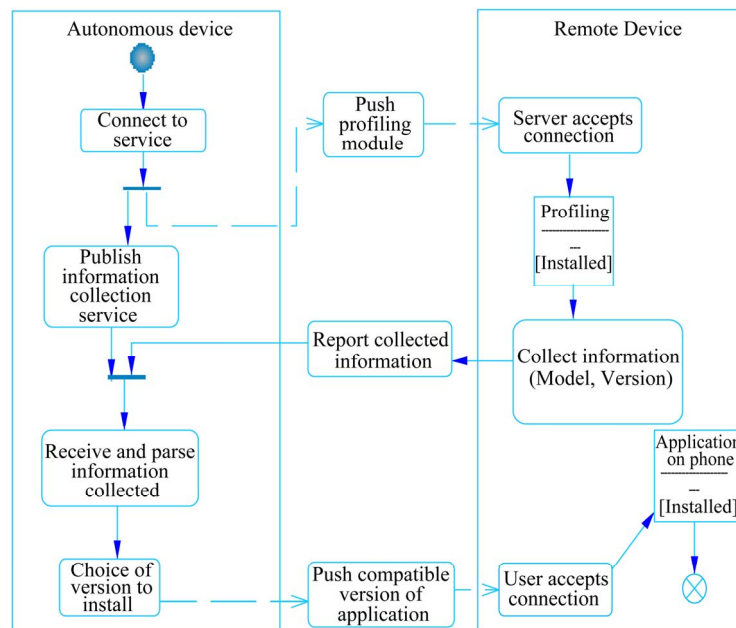**Figure 5. Activity diagram for device and service discovery.**

Figure 6. Activity diagram for profiling and provisioning of MIDlet suites.

4. Implementation [18]

The prototype is realized in **NetBeans IDE[2]**. The tools and APIs used for development are all based on the java 2 platform[3]: J2ME, Configurations[4], Profiles[5]. Additional APIs and tools comprise: **Javax.blueooth API** (used for communication between Bluetooth enabled devices), Java.obex API (used for exchange of objects between Bluetooth enabled devices), **Kuix** (a formatting tool used for decoration of mobile applications interfaces); **J2mePolish** (an ideal framework for generating different versions of application from a single code base imbedded in xml tags.

We present below a set of events taking place at the Bluetooth emitter side when it pushes applications for advertisement to mobile phones. The **Figure 7** below firstly shows the Bluetooth inquiry process for discovery of devices around the vicinity of the autonomous device, and secondly the devices discovered after a successful inquiry process, their names and the respective icons of the type of devices is shown the screen. The **Figure 8** shows the progress of the push services discovery process, once the push services are discovered, for each of the selected device (shown left-side of the **Figure 8**), the JAR and JAD files are pushed into the inbox of the remote device. The right-side of the **Figure 8** depicts the pushing process, whereby the provisioning server pushes both JAR and JAD files into the inbox of the remote Bluetooth mobile phones.

4.1. Results Analysis: Critics and Limits

The prototype realized is advantageous for advertisement without depending on the services of any mobile operator, but is somehow has a set of drawbacks:
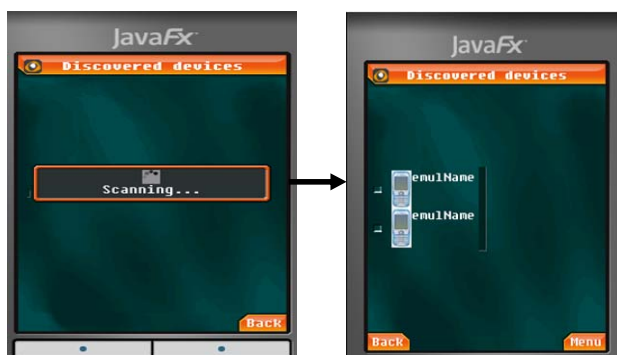

Figure 7. Devices discovery.

Figure 8. Service search and discovery.

- The applications deployed will be functional only if the targeted phone is java enabled (having the KVM).
- The discovery process is slow for emitters with low processing power. This might be disadvantageous as the client might have gotten out of proximity of the emitter.
- The prototype realized does not allow for reporting of event at the receiver side. This is an inconvenience as the client is unaware of what is about happening at is happening on his mobile phone.
- The painful process of pairing further slows down the system. This is because at each initial discovery, the client mobile phone will challenge the emitter to authenticate before pushing applications into its inbox.

5. Conclusion and Futures Works

In this paper, we described a Bluetooth-based method of propagating information within short range without depending on any mobile operator. This means of propagation is used for short range advertisement, like proximity marketing, publishing information on campus and others, just to name a few. It is also a cheap mean for propagating information to clients who are in possession of Bluetooth compatible mobile phones and thus providing a cheap and almost zero cost, means of wireless communication on mobile devices.

To render the system more efficient and increase the number of targeted clients, the researchers, interested in this project will envisage the following:

- **Study and implement the concept of Bluejacking** [7]: this is a means of escaping the Bluetooth link manager. With this concept implemented, there will be no need for the user to enter PIN codes (pairing) before exchange of data. As such the pushing process will be automatic without the awareness of the user of the mobile phone (note that it is not a thread to Bluetooth, as it only pushes objects into the inbox

---

[2]http://netbeans.org/index.html, September 2010.
[3]http://java.sun.com/javaone/online, September 2010.
[4]A configuration is a Java virtual machine plus a minimal set of class libraries (http://download.oracle.com/javame/).
[5]Define the higher-level APIs for providing the capabilities for a specific market or industry. Examples of profiles are mobile information device profiles designed for devices like mobile phones, pagers etc.

and not accessing data of the phone).

- **Render the system KVM independent**: only phones with having the k-virtual machine will have deploying applications functional. We can remove this limitation to KMV by pushing Vcards, Vmessages (which are SMS supported by all phones) for advertisement [7].
- **Use MIDP 3.0**: still under realization, this version of the mobile information device profile adds a set of functionalities that may be advantageous to our application. Functionalities like autoStart () will allows applications to auto launch themselves after being deployed and not the user taking the initiative to install the application [7] [19].
- **Implement other Bluetooth GOEP**: like the synchronization profile, imaging profile, printing profile etc.

6. Acknowledgements

Our special thanks go to Professor Jean-Claude Derniame for the time taken to review this paper.

REFERENCES

[1]   H. Labiod, H. Afifi and C. De Santis, "Wi-Fi, Bluetooth, Zigbee and Wimax," Springer, Berlin, 2007. doi:10.1007/978-1-4020-5397-9

[2]   M. B. Shoemake, "Wi-Fi (IEEE 802.11b) and Bluetooth Coexistence Issues," *White Paper*, Texas Instruments, 2001.

[3]   H. P. Lin, *et al*., "Integrating BLIP into Location-Aware System: A Service-Oriented Method," 2010. https://blog.itu.dk/KF12-F2010/files/2010/02/airnotes-with-blip.pdf

[4]   BLIP Systems A/S, BlipNet Technical White Paper, 2005. http://docs.docstoc.com/orig/460324/55d2e827-957b-4bb5-9576-fa0d3dc8388d.pdf

[5]   K. Topley, "The Mobile Information Device Profile and MIDlets," *Part 5 in "J2ME in a Nutshell"*, 2002. http://onjava.com/pub/a/onjava/excerpt/j2menut_3/index5.html

[6]   Nokia, "Settings for OTA Download of MIDlets," 2002. http://sw.nokia.com/id/9eb5643d-433a-4ee2-956b-ed0f5fb6ced9/Settings_for_OTA_Download_of_MIDlets_v1_0.pdf

[7]   D. Kammer, G. Mc Nutt and B. Senene, "Bluetooth Application Developpers Guide: The Short Range Interconect Solution," Syngress, Rockland, 2002.

[8]   G. Legg, "ZigBee: Wireless Technology for Low-Power Sensor Networks," 2007. http://www.eetimes.com/design/communications-design/4017853/ZigBee-Wireless-Technology-for-Low-Power-Sensor-Networks

[9]   S. A. Huang and R. Larry, "Bluetooth Essentials for Programmers," MIT, Cambridge University Press, New York, 2007. doi:10.1017/CBO9780511546976

[10] C. B. Kuma, T. Thompson and P. J. Kline, "Bluetooth Application Programming Using the Java API," Motorola, Inc., Ed., Morgan Kaufmann Publisher, San Francisco, 2003.

[11] B. Hopkins and A. Ranjith, "Bluetooth for Java," Apress, New York, 2003.

[12] M. Galeev, "Home Networking with Zigbee," 2006. http://www.embedded.com/showArticle.jhtml?articleID=18902431.

[13] S. Farahani, "ZigBee Wireless Networks and Transceivers," Newnes & Elsevier, 2008.

[14] A. N. Klingsheim, "J2ME Bluetooth Programming," Master's Thesis, Universitas Bergensis, Bergen, 2004.

[15] T. Thompson, "Programming Bluetooth Devices with J2ME," Rhm Ed., Morgan Kaufmann Publishers, Inc., San Francisco, 2005.

[16] Sun, "Using the Java API for Bluetooth Wireless Technology," 16 May 2010. http://www.developers.sun.com

[17] A. Davydov, "Provisioning of MIDlets," 2003. https://oa. doria.fi/handle/10024/35517

[18] K. A. Formenah, "Design and Implementation of a Mobile Application for Provisioning Applications Using Bluetooth Wireless Technology," Engineer Master of Thesis, National Advanced School of Engineering, University of Yaounde I, Cameroon, 2010.

[19] C. Hayma, "Different Approaches to Managing Fragmentation," 30 April 2010. http://www.comp.nus.edu.sg