Abstract. Mobile phone usage has become a standard practice in today’s society. With the growing usage of mobile Internet access, many new types of services have appeared that make use of the user’s mobility, such as mobile banking, instant messaging and so on. A class of services that use the user’s location as criteria – location-based services (LBS) – is of particular interest, since it allows the provision of services with information on the users vicinity. These services can be extended with rich multimedia content delivered in a push way, providing a dynamic solution for mobile marketing purposes. This work introduces a platform for housing mobile multimedia push location services with programmable logic, focusing on scalability, accessibility and flexibility of components, allowing its usage with various location systems and communication interfaces. It uses the innovative Parlay X technology, a web-services layer over a mobile network operator’s IMS architecture, to solve both the location and content push issues.

Palavras-Chave: LBS (Location Based Services), multimedia, mobile, push services, Parlay X

1. Introduction

The growth of Internet usage has made introduced a number of ubiquitous services, such as instant messaging, e-mail and social networking. Today, Internet access is possible anywhere, through current-generation mobile networks using laptops and mobile devices such as the mobile phone and smartphone. The mobility provided by these devices, in conjunction with the possibility of Internet access, as yet again caused the introduction many new types of mobile Internet services.

One such type of services takes the end-user’s geographical location to provide custom information. These are known as location-based services (LBS) [1]. The knowledge of the user’s location is important in enabling navigation services and obtaining information about events and places in the user’s vicinity. These are, however, services that have to be requested – or pulled – by the user, much like watching a weather forecast in a web page.

A more dynamic approach would be to send content to a user whenever he enters some geographical area, pushing the content directly for the user’s mobile terminal. These types of location-based push services are especially important in areas such as
mobile marketing and other information broadcast scenarios, including traffic information, tour-guides and wide-area emergency broadcasts.

The hardware capabilities of newer model mobile phones and the speed of today’s mobile network infrastructure enable services with rich multimedia content. This is of particular importance to mobile marketing and the services described above, in which the user’s satisfaction is connected with sensory stimulation and the delivery of highly customized content.

Most solutions for LBS push services do not take into account the possibility of delivering multimedia content, or are limited in the type of provided services, focusing on information, marketing or navigation only. The most widely-used mobile marketing location-based solutions also rely on the installation of proximity location detection mechanisms and the usage of push delivery protocols with low acceptance.

The limitations of existing technologies drive the purpose of this work, which is the development of a platform that is capable of containing programmable location-based service definitions with the possibility of pushing different types of multimedia content to the end-user’s mobile terminal, based on the user’s location, context information and preferences. As such, the following design goals are identified:

- **Accessibility** – The platform must enable the delivery of content to a wide range of mobile terminals, through the usage of standard technologies, and cover the widest area possible.
- **Flexibility** – The platform should support new types of location-based push services. It should also be possible to easily modify the platform in order to use new content delivery protocols and location techniques.
- **Scalability** – The growing number of users and services must be taken into account, and the platform should be able to be distributed and replicated in order to enhance system performance in such scenarios.

2. Related Work

The first step in developing the proposed platform is defining the main requirements of a LBS solution. These are:

- Mobile terminal – Where the service’s content will be viewed
- Mobile network connection – Through which the content will be delivered.
- Service provider – From which the content will be sent.
- Location system – The means by which the user will be located.

The service and content provider is the platform to be developed. The mobile terminal of choice is the mobile phone, for its wide acceptance and usage by a high number of end-users. As such, the mobile communication network of choice is the mobile operator’s network, which is the channel through which the content will be delivered to the end-user.

It is necessary to define which protocols are used for delivering multimedia content in a push way and the location system that should be used to track the phone’s location. The technologies that solve these two problems are presented in this chapter.
2.1. Location Technologies

2.1.1. Local-scale location systems

There are a number of systems that are able to determine the position of a user inside a building or some restricted space. These are known as local-scale location systems.

Portable devices can be used to emit infrared or ultrasound waves that are detected by fixed receivers, which in turn determine the position of the devices. These systems – Active Badge and Active Bat – are cost-effective, but require the installation of a dedicated receiver infrastructure and are limited to interior spaces. These technologies are also not widely deployed on mobile phones.

WiFi can also serve as a location technology by using signal-strength to calculate a terminal’s position. This is an affordable solution, since WiFi is deployed in many public areas. However, to achieve optimal precision, the density of WiFi antennas must be relatively high and the position of each antenna must be fixed and well-known, which if often not the case.

2.1.2. Global-scale location systems

In contrast with local-scale systems, global-scale location technologies allow the tracking of a user’s location anywhere on the Globe. The most widely accepted technology in this field is the GPS, which uses satellite-based location.

A GPS receiver must know the position of at least three different satellites in order to calculate the position of the user through triangulation [2]. The system is designed so that at least five satellites are visible at each instant, anywhere on the Globe.

This technology can only determine the position of a user in an exterior space, since the GPS signal is easily blocked by walls and suffers from atmospheric attenuation and reflection. A-GPS is designed to minimize these flaws by using mobile network-based positioning as an auxiliary location method, achieving a higher accuracy and allowing the system to work in interior spaces.

Most mobile phones do not have GPS or A-GPS, compromising the accessibility of a solution that uses them exclusively for mobile position determination.

2.1.3. Mobile Network-based location

Mobile operator networks are composed of a series of fixed antennas which enable mobile phones to communicate with each other through GSM. The area served by one such antenna is called a Cell, having a unique identifier within the network.

It is possible to obtain the position of a mobile phone connected to the GSM network through various techniques, the simplest being Cell ID [2], which calculates the approximate position of the user by knowing the position of the cell the mobile phone is connected to. With more cells, other location techniques are possible, such as TDOA (Time Difference of Arrival) and AOA (Angle of Arrival), which calculate the position using information about the time or the angle of arrival of a signal that reaches the cell’s base station.
All these techniques have the advantage of using the existing mobile network infrastructure, making them the most available solution for location. The disadvantage lies in the precision of these techniques, which is directly proportional to the number of cells a mobile phone is connected to, and their density. In worst-case scenarios, the precision error can be as great as 25 kilometers, in rural areas, with an average of 200 meters in urban areas with a large cell density.

2.2. Content Push Technologies

2.2.1. Proprietary Application

The development of a proprietary application and underlying communication protocol solves the content push delivery issue on mobile phones. This requires that the user installs a J2ME application on the mobile phone which is to be run as a daemon, continuously maintaining an open connection to a server.

This solution is however undesirable, since it will cause the unnecessary consumption of a mobile phone’s CPU cycles and additional battery drainage. The acceptance of such an application is also questionable, as well as the necessity of developing a different application for every type of mobile phone marketed (which can be mitigated by using cross-platform solutions like J2ME Polish and Phonegap).

2.2.2. MMS

MMS is mobile messaging to send multimedia content between end-users. This solution is a successor to SMS messaging, removing the 160 character restriction and allowing multimedia content.

This technology is push-based, like SMS, but requires that the recipient’s mobile phone be correctly configured. Also, the mobile network operator may use unicast to deliver a MMS to multiple recipients, compromising the scalability of the system for mass mobile marketing campaigns.

2.2.3. WAP-Push

WAP-Push allows arbitrary content to be pushed to a recipient’s mobile phone, via a URL of the content that is sent to the user via SMS. The URL is then opened on the phone’s built-in WAP browser, displaying the content.

The disadvantage of this solution is that the content must be pulled by the user, since only a URL is sent. This is called push-to-pull. It however allows video streaming, by sending the user a RTSP streaming URL, which will be opened by the phone’s browser and relayed to the phone’s media player, if the video format is supported.
2.2.4. MBMS

MBMS is a protocol that enables the broadcast of multimedia content to multiple mobile phones in a scalable way, providing two working modes – download and streaming. The first is used to download arbitrary files, using the FLUTE protocol. The latter allows broadcast of video and audio streams, encoded in H.264 and MPEG4/AAC respectively, via RTSP [3].

The main disadvantage of MBMS is that most mobile phones do not implement a MBMS client, requiring the installation of a third-party application.

2.3. Location and Push-delivery Solutions

2.3.1. Bluetooth

The Bluetooth technology solves both the location and content-delivery problems. Like WiFi, Bluetooth is a radiofrequency technology that has a range of approximately 10 meters. Signal strength-based techniques can be applied to determine the position of a mobile phone connected to a Bluetooth access point. Content delivery is also possible, through Bluetooth file sharing capabilities.

However, in some cases, this technology requires that both the access point and mobile phone be paired beforehand, and that the Bluetooth radio be enabled. This is mostly a manual operation, requiring some degree of user interaction.

Also, unlike the mobile operator network, Bluetooth access points must be installed where the content delivery services are to be delivered, not leveraging the usage of an existent network infrastructure.

2.3.2. Parlay X and OneAPI

Parlay X [4] is an innovative and emerging technology that exposes functions of a mobile network operator’s IMS architecture through a web-services layer. With the current specification, 20 APIs are exposed. Notable APIs are Terminal Location, Multimedia Messaging, Terminal Status, Multimedia Streaming Control and Multimedia Multicast Session management. OneAPI is a work-in-progress Parlay X derivative implementation.

Through these APIs, this technology enables the delivery of multimedia content in a push way, using SMS, MMS, WAP-Push and a control layer over MBMS, providing a high degree of flexibility in content delivery options. It also serves as a location system by using the Terminal Location API, which in worst-case scenarios uses the Cell ID network-based location system to determine a user’s position. Finally, it’s possible to obtain information on a user’s context by using APIs such as Terminal Status.

A solution leveraging Parlay X accomplishes the design goals of Accessibility - through the usage of accessible technology, the mobile operator network and standard content delivery protocols – Flexibility – by providing various ways of delivering multimedia content and by being in constant evolution – and Scalability – by
providing access to the scalable MBMS protocol for content broadcast and wide area coverage provided by the mobile operator network.

3. Architecture

Having analyzed the technologies available for content delivery and mobile location, a solution was designed to enable the delivery of mobile location-based push services with multimedia content. Figure 1 represents the component diagram for the proposed solution.

![Solution Architecture Diagram](image)

**Figure 1 - Solution architecture**

3.1. Interface Component

The mobile operator network is transparent to the solution, housing the Parlay X gateway. The Interface component accesses the functions provided by the Parlay X gateway, organizing them in three modules – Location, Data Transfer and Context.

The Location module provides access to the terminal location Parlay X services, also receiving notifications when users enter a geographical area. Data transfer has functions for sending content via SMS, MMS, WAP-Push and e-mail. Finally, the context module accesses Parlay X context retrieval functions – Terminal status and Presence.
3.2. Presentation Component

This component houses the graphical user interfaces that will be used to control the system. The User Portal allows end-users to change their preferences and subscribe or block services. The Administration Portal is used to create services, geographical areas, users and contents. Finally, the Service Provider Portal allows service and content providers to register services and deploy content to the system.

3.3. Logic Component

Being the central component, it intermediates communication between the presentation and data persistence components, by exporting web-service interfaces – Data Interface. It is also responsible for processing the event when a user enters a geographical area – Event Manager –, choosing the appropriate service – Profile Manager – and sending its content to the user’s mobile phone – Service Logic Processor.

3.4. Persistence Component

The persistence component mediates the access to the system’s databases, performing some of the logic that otherwise would be in stored procedures or triggers. Three types of subcomponents are designed, each one responsible for one database, for users, services and geographical areas and content.

4. Implementation

The architecture described in the previous chapter is a materialization of the system requirements. Such architecture was implemented in three steps, for each main component – Persistence, Logic and Interfaces and Presentation.

4.1. Persistence

The data access layer of the persistence components was programmed using Hibernate, allowing database independence and promoting flexibility and future scalability when using distributed databases. It exports interfaces to access data through SOAP over HTTP web-services, being deployed as three distinct web applications – User Server, Service Server and Content Server – which provide some degree of logic and transformation of database registries into appropriate views.

Since the three databases share some keys, an automatic daemon was deployed to assure that key consistency is maintained across components.
4.2. Logic and Interfaces

The logic component is implemented as an EJB3 bean deployed in a jBoss 5.1.0 GA application server. It maintains a singleton instance for each of the Interface modules defined in the architecture, which are interchangeable with other versions that comply with the same interface. The communication with external systems is possible through the use of a connector library that uses the Ericsson ParlayX SDK functions.

The event manager is implemented by a thread pool. Every time a user enters an area, a thread receives an area ID and user MSISDN, passing them to a new thread that implements the functions of the profile manager.

Profile manager threads use an algorithm for choosing the most appropriate service for the user, based on the number of matching preferences between them, the service’s schedule and whether the user has subscribed or blocked a specific service.

After choosing a service, the service logic is extracted and executed. Service logic is defined in a script by means of a formal language developed for this purpose – LobsterScript. This script contains functions that will effectively send the content through the Interfaces module.

The logic component also, periodically, synchronizes users and areas on the ParlayX gateway, since they have an expiry date, for creating new area entry notification callbacks. This is performed by removing all registered callbacks and registering them again with all area IDs and user MSISDNs in the system.

4.3. Presentation

The presentation component was implemented as two web applications written using SmartGWT (a derivative of GWT) over Java – User Portal and Administration Portal. It communicates with the logic component through SOAP over HTTP web-services.

The administration portal encompasses both the service provider and administration modules defined in the architecture. Its user interface is implemented following the desktop paradigm, with functionalities present as icons on the user’s desktop. These functionalities, known as virtual applications, are four in number – user manager, service manager, area manager and content manager. Each virtual application has different usage privileges, depending on the user logged in the administration portal, and provides means of managing users, creating services, defining geographical areas on a map and referencing remote multimedia content.

The user portal is another web application designed to be integrated with the Facebook social network. This integration provides an efficient authentication method, not requiring the user’s explicit registration and login, and context information pulled directly from the user’s profile in the social network.
5. Solution Evaluation

In order to determine the functionality, usability and performance of the solution, two types of tests were performed – Functionality Tests and Non-functionality tests.

5.1. Functionality Tests

For evaluating the solution’s functionality, component integration and system tests were performed. These have the purpose of verifying that all components interact correctly with each other and that the system as a whole works as expected. Integration tests were performed in growing levels of interaction, from the reception of a notification from the Parlay X gateway all the way to the reception of a service’s content in the end-user’s mobile phone, through SMS, MMS and WAP-Push, which culminated in a whole-system test. All components were tested to be working as expected, as was the system as a whole.

5.2. Non-functionality Tests

For testing non-functionality, two types of test were conducted – performance tests and usability tests. Both the time it takes for a user’s location to be detected and the time it takes for the content to be delivered were tested for performance, with a worst-case scenario of 1190 and 1436 milliseconds for both operations, which means that a service will be delivered in, at most, three seconds approximately.

Usability tests were performed by user observation, while operating the user portal, and an enquiry, using a test population of 25 users. Usability proved to be within acceptable levels, with good ease-of-use and coherence results and average appearance results.

6. Conclusion

The motivation for building a mobile push location-based service (LBS) delivery platform was presented in the beginning of this article, providing a context on the evolution of mobile platforms as new service targets, and detailing the design goals of such a platform – scalability, flexibility and accessibility.

In the related work, two main areas of research were identified for developing a LBS platform – location system and content push delivery method. A series of technologies were identified, with Parlay X being the best choice for both areas.

An architecture was designed, with three main, independent and distributable, components to develop – Interfaces, providing access to Parlay X services for mobile terminal location, content delivery and context retrieval – Presentation, consisting of a series of user portals – Logic, mediating presentation and data as well as performing computations for service delivery – and Persistence, with database access components which encapsulate it.
For each of these main components, the implementation details and technology used were explained in the subsequent section, culminating with functionality and non-functionality tests in the last section. Test results validate the solution’s requirements, also with good performance and usability.

The solution presented using Parlay X thus achieves all design goals, while being innovative by the use of that technology. It is scalable through the distribution of the architecture components and possible replication, accessible, through the usage of an accessible technology – Parlay X – and flexible by providing service programmability and interchangeable location and push content delivery interfaces.

6.1. Future work

Despite being innovative and fulfilling all the functional and non-functional requirements defined, there are some enhancements that can be left as future work.

Security is one of the first concerns, with the necessity of implementing authentication in the component API, to avoid external unauthorized access.

Integration of future OneAPI also requires some changes in the way location notifications are created and handled. It is necessary to define a scalable way to implement the notification mechanisms of Parlay X on the solution’s side.

Since the BM-SC component required for MBMS was unavailable, integration with this protocol was impossible. This should be leveraged in future iterations of the platform, since it is an emerging multimedia protocol.

Finally, with the ability to have programmable service logic, a way of providing user interaction with services should be researched. This would allow the existence of interactive services, such as games and quizzes.

Bibliography