A Location-Based Transactional Download Service of Contextualized Multimedia Content for Mobile Clients

Pablo Fernandez, Asier Perallos, Nekane Sainz, and Roberto Carballedo

DeustoTech - University of Deusto, Avenida de las Universidades 24,
48007 Bilbao, Spain
{pablo.fernandez, perallos, nekane.sainz,
roberto.carballedo}@deusto.es

Abstract. This paper explores the new opportunities offered by the emerging technologies of the last generation of mobile phones. Thanks to features like GPS facilities installed in the mobile terminals new value added services can be developed to offer the user the more suitable multimedia content depending on parameters like user's location and preferences. We describe the development of a contextualized and personalized multimedia content delivery platform using transactional communications for mobile terminals. Furthermore an actual test route has been made to proof the successful working of the platform.

Keywords: pervasive information system, context-aware, location-based, download service, transactional, multimedia contents, mobile device.

1 Introduction

Since the release of the first portable audio player in 1979, the Sony Walkman, the portable entertainment has evolved and we are in a moment of history where the intelligent mobile phone devices are the preferred portable entertainment devices, their prices have decreased and mobile Internet connection rates have became more accessible [1]. New terminals are equipped with multimedia capabilities and are easier to handle. This mobile device evolution motivates the change from voice-only communications towards mobile multimedia contents and the offer of new value added services in industries like entertainment and tourism.

Furthermore, GPS devices are being integrated in most of the modern mobile terminals [2], and thanks to this a lot of new intelligent value-added services that currently are not widely exploited can be offered. One of these services is the contextualized multimedia content delivery to mobility environment, attending to the client geographical position and personal preferences, as it is described in [3].

The aim of the work described in this paper is just to explore such possibilities through the development of a pervasive platform for providing contents on demand using mobile terminals in a mobility environment (transportation scenario) where the context of the user is considered (location, desired destination, preferences, etc.) in order to know which contents have to be provided in a proactive way.

2 Functionality

To review the system features and functionality we should first identify two main actors in the platform: *mobile clients* who discover new content and services when they are in motion; and *publishers* which contribute to the system with geo-referenced and subject-based contents. The platform is based in subscription model where a mobile client is offered with new contents when he is located in a specific location and he can subscribe to that content so the information is automatically downloaded to the mobile client.

Thanks to the functionality of the system, a tourist is able to arrive in a city he does not know and download the information he is interested in and it is near his location. Related work in tourism scenarios is described in [4] and [5]. It does not matter if he is driving or walking, the system has two different profiles not to bother the user with no necessary notifications. The application can be used in different mobile terminals because is compatible with a large number of devices and the system is able to personalize the multimedia content to the specific features of the user`s mobile phone. So if the tourist unfortunately loses his phone, he will be able to get a new one and start using the application as before. Furthermore, if the user receives a phone call in the middle of the downloading process of a multimedia content, the platform can resume that download after the phone call, thanks to the transactional downloading feature.

The technical functionalities of the system which we consider most innovative are briefly explained below:

- *Multimedia File Re-encoding for Personalized Mobile Content.* The platform is going to offer multimedia files to the clients, and they will usually be of a major size. In order to save data transmission traffic and resources, the platform is able to serve a personalized multimedia file to the client invoking its petition. So the platform recognizes the mobile phone specific features such as display size and codec playing ability, and sends him a re-encoded multimedia file with the display resolution and a suitable codec. This is done thanks to a large mobile phone database where the devices have associated a unique User-Agent with their specifications. When a client invokes a petition, their User-Agent is embedded in the HTTP request so the platform can compare it in its database and send the appropriate content file. If the User-Agent of the mobile phone is not found, the platform is able to read raw data of the display size of the device instead of looking in the database.
- *Large Compatibility with Mobile Devices.* There has been developed a code library in several programming languages like Java ME, Java, Windows Mobile, so the client can be compatible with a large amount of devices, portables or not. However the library can be easily ported to the new generation operative systems of smart phones like Iphone's OS or Google's Android. Furthermore thanks to the personalized mobile content, the platform ensures that every mobile phone is going to receive a suitable media file to its specifications, so every terminal is compatible with the system.

- *Geo-Referenced and Subject-based Contents.* Using a user-friendly interface the content publisher is able to upload new content in an easy way. With an interactive map, the location coordinates and the activation range are automatically assigned, and then the publisher is able to insert keywords or subjects to describe the content. In the client-side the user is able to configure his favorite subjects and keywords, as well as the range within it is going to receive new content offers.
- *Load Balancing System and Distributed File Storage Architecture.* The platform architecture will be explained in further sections of this document but it is important to notice that multimedia files are stored in several servers so when a client invokes a download petition the platform can choose the proper server to download from, attending to parameters like, bandwidth, memory or CPU usage of the servers. Thanks to this feature, the platform can maintain a load balanced environment to try to always offer the best quality service.
- *Transactional Downloading.* One major disadvantage of using mobile phones with over the air internet connection is the low speed of the data transmission, and the poor high bandwidth coverage. Thanks to transactional downloading, no matter if the connection is interrupted in the middle of a download, because you can resume the download later in the exact point you were the last time.

3 Platform Architecture

In the architecture schema we can identify four key components which are shown in Fig.1. Their responsibilities are briefly explained below.

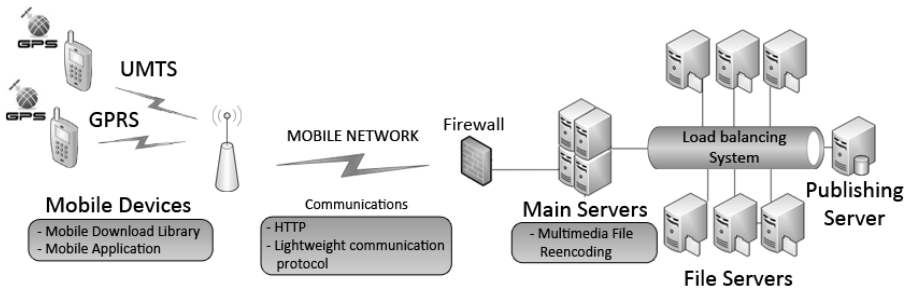

Fig. 1. Network architecture and key components

1. *Main Server.* It manages the user authentication, new content discovery services and business logic like enrooting the download request to a specific file server.
2. *File Server.* It contains all the content files of the system, it is a lightweight server that just has one task, replying requests from the Main Server.
3. *Content Publishing Server.* It runs in a web server where the publishing portal is hosted, when a new file is published, this is automatically replicated to all the file servers.
4. *Mobile Devices.* They are mobile terminals with GPS capabilities where the client application is running to retrieve the geo-referenced information.

The system is based on a centralized architecture. In this way the Main Server acts as a 'gateway' which receives all the requests made by the mobile phones. Thus, centralizing the connections grants the next features: (1) precise control of all the requests; (2) just one central point to ease the charging tasks; (3) security.

The mobile-server communications are established with a client-server topology, due to the fact that its transactional downloading and lightweight communications can be handled flawlessly by most of the mobile devices in the market.

4 System Design and Implementation

4.1 Communications Module

This module has the responsibility of implementing a lightweight communication protocol which has been designed in order to manage the interaction between the mobile device and the server in a secure and fast way. This lightweight protocol has several simple messages and can handle the next functionality:

With the porpoise to obtain a secure and fast communication, we have designed a lightweight protocol of simple messages that can handle the next functionality:

- Discover the available content for a specific user
- Send and receive GPS and location data
- Identify the mobile device and its specifications (because of the platform is able to provide personalized content for a given mobile device specifications, the protocol must be able to send the necessary information to the server).
- Transactional file sending

In order to free the main server from handling large volumes of data, all the information related to the status of a download, like the chunks that have not been downloaded, is stored in the mobile device, so we can ensure the main server has the minimum load to attend all the requests made by the mobile devices.

The basic life cycle of the interaction between the mobile device and the server is guided by the next five actions: (1) Negotiation → (2) User Identification → (3) File List Request → (4) File Information Request → (5) Chunk Request.

In addition to the previous cycle, there is a timer that is responsible of sending the location coordinates to the server using the GPS device installed in the mobile device. The time between each petition is configurable so the user can save battery life.

As we previously explained, one of the main features of the platform is the ability to offer personalized content according to the mobile device model and specifications. In order to make this identification, the mobile device sends its User-Agent. This is a unique string that identifies the mobile phone. But there may be situations where the mobile phone cannot be identified. In these cases the platform has an alternative mechanism to offer custom multimedia content: the mobile phone is able to send its screen resolution, so the platform can offer him the more suitable version of the file.

The transport protocol used for the communications is http because it is a standard, it has secure connections tools, and it can work through firewalls and proxies. Above

the http protocol, the application layer protocol has been developed using custom Servlets and EJBs for attending the requests from the mobile client. As we have mentioned before, we have several messages that are part of our lightweight communication protocol. For attending these we have developed one Servlet per message that has to be sent. Working with Servlets is an easy task that involves using standard http connections that have implementations in all programming languages.

Web Services it is another good choice because they have many advantages like the standardization of the communications and the easy development process, but it has a major fault; it works with XML files, and mobile devices do not behave well with them because they take a lot of process time in the mobile terminal.

4.2 Mobile Device Module

The mobile device module is made up of the mobile download library and the user mobile applications. In this section we are going to describe such components.

4.2.1 Mobile Download Library for Mobile Devices

In order to make the platform usable for a large number of devices, a code library that can be used by any third party client application has been developed. This way, anyone can make his own client application and use the entire download platform. This library has been coded in three different programming languages, J2ME, J2SE and Windows Mobile in order to the major part of the mobile devices in the market can be supported. The functionality of this library is summarized in these four tasks:

1. *Server Communication.* As we mentioned in the previous section, in this library is where the lightweight communication protocol is implemented. Thanks to this, the client application can 'talk' with the server, authenticate, request new files and download chunks.
2. *Download Management.* This functionality is responsible of all the transactional downloading tasks. Including error handling and stopped downloads resuming.
3. *GPS and Location Data Manager.* The mobile module must be able to send periodically the GPS data. In this information are included the longitude and latitude coordinates as well as the direction towards the mobile device is moving. This periodical dispatch can be easily configured to save battery life. In this case the send rate will be lower and the accuracy of the position will be less precise.
4. *File Storage.* This module must first be able to store the chunks downloaded in a temporary folder, as well as the file information, and secondly it has to rebuild the file and store it in its final destination when all the chunks have been downloaded.

These functionalities are implemented in the code library and to use it, the client application must implement some interfaces defined by the library. These interfaces are responsible for alerting the application when an error has occurred, notifying any rare situation, logging events to a log file and throwing custom exceptions.

4.2.2 Mobile Application

Using the mobile download library, we have developed a fully functional J2ME and Windows Mobile client applications, which have been tested by real users. As we told in the previously the client has two preconfigured working profiles: Drive and Walk.

- In *Drive* mode, the application is continuously sending the GPS data to the server to ensure the high accuracy of the position. Due to this fact, is recommended to have the mobile device plugged in to the power line because this is a battery consuming profile. In this mode the user will not receive any notification; offers that are near his location and matches with his preferences will be automatically subscribed so the content is downloaded immediately.
- In *Walk* mode, the GPS data sending rate is decreased in order to save battery life. The position accuracy is lower but this is not a vital factor due to when a user is walking, his speed is not fast enough to lose any offer. When the user receives a notification, he can see a detailed view of the offer, including its description, keywords and position in a map. If he is interested in the offer he just has to subscribe to start downloading the multimedia content.

Besides this two working modes, with this application the final user can do all the tasks described in the section 2 of this paper. It means: (1) automatic transactional download of contents; (2) network traffic watch and download status; (3) downloaded contents management; (4) play downloaded contents: mp3, videos, and pictures; (5) user profile configuration: credentials, keyword interests, range offer activation.

4.3 Server Module

For all the three types of servers the technology chosen has been JBoss AS. This application server has already a web server included to support the EJBs and the Servlets needed to communicate with the mobile library. Moreover it is able to work in cluster mode in a very easy and secure way. This means one single application can be deployed and shared by several servers to always ensure the optimal performance of the application. Thanks to the cluster mode, the platform is easily scalable because it only has to add more servers in the same cluster to start working together.

Although the three types of servers use JBoss, each one is focused on specific tasks. Main Server implements all the business logic to receive the petitions from the mobile devices; this logic is supported by EJBs and Servlets. File Servers have the only task to listen requests from the main server. These requests are made over JBoss Remoting, which is a substrate of JBoss AS and it is a lightweight communications framework that is able to use http connections. And finally Content Publishing Server is a web portal hosted in the JBoss Web Server responsible for the publishing of new content from a private or public provider. In this Web Portal the publisher can manage his offers and topics and upload multimedia content. To assign GPS coordinates and activation range of the offer we have designed an interactive application over Google Maps to manage the content in a very intuitive way. We just has to point in a map and he will receive a visual feedback of the location and activation range of his offer.

5 Testing and Results

For testing purposes we populated the platform with four different offers located in the city of Bilbao (Spain). Their GPS coordinates and information are presented in the Table 1. The test consisted in a route through the city with the mobile client application running in a "*HTC HD2*" mobile passing over all the activation ranges.

Table 1. Offer locations and information

Offer Name	Longitude	Latitude	Activation Range
Museo Guggenheim	-2.934250831604	43.2686437747697	400metres
Museo Arqueologico,	-2.92824268341064	43.2628937385262	450metres
Polideportivo Deusto	-2.94784426689148	43.2684719065613	500metres
Gran Hotel Domine	-2.93384313583374	43.2677375551133	200metres

We started our test route and put the application in Walk mode and we started walking. When we reached the first activation range, a notification appeared telling us there was an available offer in our position that matched our preferences. We subscribed to that offer and continue walking. The content of the first offer named *"Polideportivo de Deusto"* was automatically downloaded including three pictures in jpeg format and a pdf file with the prices of the establishment.

We continued our test route and put the application in Drive mode so it will not notify us of anything. We passed through the rest of the three activation ranges of the offers and the application automatically subscribed and started downloading the contents. In the middle of our test we stopped the application to test the transactional downloading process. When we restarted the mobile application, the download process resumed in the point it was before the restart. We got to the end of our test route and checked the application; it correctly had downloaded all the multimedia content of the offers, including two video files, one audio file, and several pdf files with information about the offers. This test was executed using the 3G mobile network for the download traffic and it successfully worked with an average download speed of 1832 kbps. The results of the test can be seen in the Fig. 2.

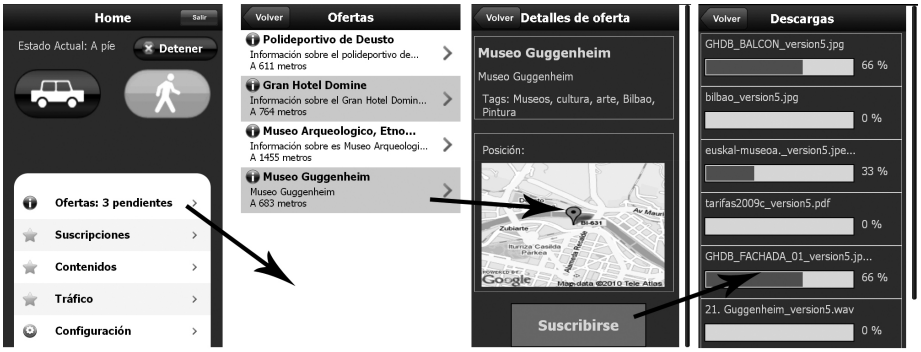

Fig. 2. Mobile application interface and interaction

After finishing the experiment we can conclude we obtain successful and positive results in all the areas of the platform: the GPS data and location sending from the mobile client; new offer and content discovery in Walk mode; automatic subscribing and downloading in Drive mode, transactional downloading and personalized multimedia content based on mobile device specifications.

6 Conclusion and Future Work

In order to study some new possibilities that the mobile multimedia technologies are able to offer, we have successfully developed a platform capable of combine these ones with the advances in location solutions. The result of the work is a successful development of an innovative platform for the proactive provision (using on mobile devices) of multimedia contents which are contextualized to the user profile (location, desired destination, preferences, etc.). Therefore, it includes very advanced characteristics such as multimedia file re-encoding for personalize mobile contents, large compatibility of client application with mobile devices, distributed storage of contents and a load balancing system.

Future work is focused on the improvement of the content contextualization by using semantic webs and ontologies as described in [6], which can interpret the user data and preferences and discover which would be the best contents for his interests. Other goal is to allow the user to participate in the platform, allowing him to publish new contextualized content with his mobile phone, and integrating the application and content with the most important social networks.

References

1. Harroud, H., Karmouch, A.: A Policy Based Context-aware Agent Framework to Support Users Mobility. Proc of the IEEE Advanced Industrial Conference on Telecommunications / Service Assurance with Partial and Intermittent Resources Conference / ELearning on Telecommunications Workshop, Lisbon, Portugal 17-20 July (2005).
2. Yu, Y-H., Kim, J., Shin, K., Jo, G-S.: Recommendation system using location-based ontology on wireless internet: An example of collective intelligence by using 'mashup' applications. Expert Systems with Applications (November 2009).
3. Vijayalakshmi, M., Kannan, A.: Proactive location-based context aware services using agents. International Journal of Mobile Communications (2009)
4. Martínez, D., Ruíz, N., Vera, P.and, Alcántara L.: Multimedia platform for mobile tourist guidance and services in the cities of Úbeda and Baeza (Spain). International Conference on Wireless Informatio Networks and Systems. Winsys 2009, Italy, Milan 7-10 July (2009).
5. Kakaletris, G., Varoutas, D., Katsianis, D., Sphicopoulos, T., and Kouvas, G.: Designing and Implementing an Open Infrastructure for Location-Based, Tourism-Related Content Delivery. Wireless Personal Communications (30), pp.153–165 (2004).
6. Bellavista, P., Corradi, A., Montanari, R., Toninelli, A.: Context-aware semantic discovery for next generation mobile systems. IEEE Communications Magazine (September 2006).