An InfoStation-based Distributed mLearning System

Zhanlin Ji, Ivan Ganchev, Máirtín O’Droma
Telecommunications Research Centre (TRC), University of Limerick, Ireland
{Zhanlin.Ji; Ivan.Ganchev; Mairtin.ODroma}@ul.ie

Abstract—Within this paper, a client-server mobile e-Learning (mLearning) system is presented, which operates within an InfoStation-based networking environment for the provision of mobile services to users under the always best connected and best served (ABC&S) paradigm. A light-weight distributed framework is utilized for service governance on the server side, along with a generic Android mobile application operating on the client side. This distributed design helps improving the system’s reliability and performance as it allows mLearning services to be deployed on different machines located in different areas.

Keywords—mLearning; InfoStations; Android; Mobile Application; Distributed Framework; ZooKeeper

I. INTRODUCTION

The mobile eLearning (mLearning) [1] system presented in this paper could be used to enhance traditional educational practices or as an additional tool to support the distance learning process. It is deployed to operate on an InfoStation-based network as a ‘anywhere-anytime-anyhow’ oriented system [2]. With a multi-agent system support [2], the requested mLearning services are personalized and customized to suit each individual mobile user who gets access to them through the ‘best’ wireless connection under the always best connected and best served (ABC&S) communication paradigm [3]. The system presents a novel infrastructural solution for mLearning service provisioning facilitated by the next-generation mobile applications.

The utilization of an InfoStation-based network architecture by the system has been described in [4]. This architecture is context-aware and agent-oriented, and consists of three tiers, as shown in Figure 1.

Figure 1. The 3-tier InfoStation-based network architecture [4].

Tier 1: User mobile devices. A personal Assistant Agent (PAA), installed on each mobile device, facilitates the initial authentication, authorization and accounting (AAA) of the user, personalizes the user requests for mLearning services (i.e., mTest, mLecture, mLibrary, etc.), and visualizes the results of the service requests’ execution.

Tier 2: InfoStations. Besides facilitating mobile users with access to the mLearning services via the ‘best’ available wireless connection, the InfoStations provide dynamic service/content adaptation by taking into account the current user/service/network context for enhanced service personalization and contextualization.

Tier 3: InfoStation Centre. It provides a high-performance distributed environment for the efficient provision of mLearning services. The InfoStation Centre (server side) controls, updates and synchronizes information across all connected InfoStations within the network. It contains also user profile managers, which record mobile user’s personal information, including user’s location, ID, test results, interests, mobile devices, etc. and service managers, which facilitate the delivery of personalized service content to users based on the current context.

The concept of this mLearning system has been discussed in detail in [5] along with corresponding implementation methods utilizing the Java Agent DEvelopment framework (JADE) [6], the Wireless Universal Resource File (WURFL) [7], and the Wireless Abstraction Library by Luca - Next Generation (WNG) [8]. A distributed framework for the provision of mLearning services has been introduced there along with its main component—the inversion of control (IoC) light-weight container.

A new version of the software architecture of this mLearning system, along with its associated client and server components, is the focus of the current paper. The paper details the re-designing of the system with a client-server design pattern. An Android mobile application was developed to operate on the client side and a distributed web application operates on the server side for improving the system performance.

II. SERVER SIDE

The server side is built with a light-weight distributed framework which allows loose coupling between the architectural components. Figure 2 depicts the multi-component architecture of the server side of the system.

The varnish cache [9] acts as a reverse proxy server for accelerating the HTTP responses. The web server runs a number of Nginx servers providing a load balancing function. A producer/consumer model serves as a distributed framework for service governance (Figure 3), which provides sync-over-async and request-response messaging functions, load-balancing/failover/clustering capabilities with a remote procedure call (RPC) mechanism, etc. A Zookeeper [10] acts as a register centre for the services. The producer registers a service to the Zookeeper, when
a consumer subscribes to a service. Then the Zookeeper notifies the consumer and the corresponding service is invoked. The leader in Figure 3 is elected by a Quorum. If the leader goes down, a new leader is elected. The Redis cluster was built with master nodes and slave nodes. The Sphinx index [11] provides a full-text search functionality for Android mobile applications. The Hadoop is used as a cloud environment for running Map/Reduce algorithms and supporting the distributed file system.

Figure 2. The Distributed Architecture of the Server Side.

III. CLIENT SIDE

The Allied Business Intelligence (ABI) Research [12] reported that the Android operating system [13] played a dominant role in the smartphone market in 2013 (with 81% share). That is why the main focus was on the designing an Android-based mobile application, which to facilitate the mobile access to mLearning services for users equipped with mobile devices, helped by a personal assistant agent (PAA).

In the PAA design, a hardware-independent uniform JADE-Android design was elaborated, based on the JADE-LEAP [14]. To achieve this, a HTTP-like request/response mechanism, together with a compact Java object serialization APIs, was developed. Figure 4 shows the class diagram design of the PAA.

III. CLIENT SIDE

The Allied Business Intelligence (ABI) Research [12] reported that the Android operating system [13] played a dominant role in the smartphone market in 2013 (with 81% share). That is why the main focus was on the designing an Android-based mobile application, which to facilitate the mobile access to mLearning services for users equipped with mobile devices, helped by a personal assistant agent (PAA).

In the PAA design, a hardware-independent uniform JADE-Android design was elaborated, based on the JADE-LEAP [14]. To achieve this, a HTTP-like request/response mechanism, together with a compact Java object serialization APIs, was developed. Figure 4 shows the class diagram design of the PAA.

III. CLIENT SIDE

The Allied Business Intelligence (ABI) Research [12] reported that the Android operating system [13] played a dominant role in the smartphone market in 2013 (with 81% share). That is why the main focus was on the designing an Android-based mobile application, which to facilitate the mobile access to mLearning services for users equipped with mobile devices, helped by a personal assistant agent (PAA).

In the PAA design, a hardware-independent uniform JADE-Android design was elaborated, based on the JADE-LEAP [14]. To achieve this, a HTTP-like request/response mechanism, together with a compact Java object serialization APIs, was developed. Figure 4 shows the class diagram design of the PAA.

In the Android application design, the source code was divided into seven Java packages, as shown in Figure 5. The model includes a number of Plain Old Java Object (POJO), i.e., course, history, video, profiles, etc. The cache provides ContentProvider functions for the Android application. The entrance of the Android application is the HomeActivity. An android.widget.TabHost container was selected to act as a window view. After the creation
of the container, the adapter and fragment components are initialized. Then the Android application returns the corresponding activity to the user. Figure 6 shows the HomeActivity time sequence diagram.

IV. RESULTS

The development of the new version of the mLearning system followed the personal software process (PSP) methodology [15]. Test-driven and feature-driven development methods were used for the PSP.  

Server Side: The performance of the service producer/consumer model was evaluated as this is the key element in the server side. A Zookeeper cluster was built, running on three XEON Intel 5520 servers - one is the leader and the other two are followers. Three different sizes of data blocks (128B, 1024B, and 4096B) were used for performing concurrent read-operations with different number of threads. The evaluation results shown in Figure 7, clearly demonstrate that the smaller the block size, the better the throughput. Also, the results suggest that using more than 10 concurrent threads is not advisable for this cluster configuration as it will lead to a significant degradation of the system throughput.

Client side: Figure 8 shows the GUI of the developed mLearning Android application. It includes three tabs:
(i) the Best tab (Figures 8a/b/c) includes a search box for performing search actions, two fragments for "best" courses (suitable for that particular user), and a list of categories. When the user selects a category, a universal image loader is invoked for caching and displaying the course’s images. After clicking on a particular course, the list of sections is displayed to the user; (ii) the Mine tab (Figure 8d) stores the user’s profiles, history, etc.; (iii) the Down tab keeps information about all files downloaded by the user, i.e., offline videos, course lectures, etc.

V. CONCLUSION

The improved design and implementation of a client-server InfoStation-based mLearning system have been sketched in this paper. On the server side, a distributed architecture has been used for service governance, based on the producer/consumer service model and utilizing a Zookeeper as a service register. On the client side, an Android mobile application and a personal assistant agent (PAA) have been designed, as representatives of the next-generation mobile applications, for the provision of highly personalized and contextualized mLearning services to users under the always best connected and best served (ABC&S) paradigm.

ACKNOWLEDGMENT

This publication has been supported by the Irish Research Council (IRC) and the Telecommunications Research Centre (TRC), University of Limerick, Ireland.

REFERENCES


