The First Android Client Application for the iLab Shared Architecture

http://dx.doi.org/10.3991/ijoe.v8i1.1946

B. Deaky[1], D.G. Zutin[2] and P. H. Bailey[3]
[1]Transilvania University of Brasov, Brasov, Romania
[2]Carinthia University of Applied Sciences, Villach, Austria
[3] Massachusetts Institute of Technology, Cambridge, MA, USA

***Abstract*—This paper presents the first Android client application developed for online laboratories based on the iLab Shared Architecture. An important challenge was to properly connect to the ISA Service Broker, because its current version was developed with browser-based client applications in mind.The application was successfully tested on a few real-world mobile devices and the experience gained represents the basis for future changes in the Service Broker and for future teleengineering applications that involve Android.**

***Index Terms*—remote engineering, online laboratories, Android.**

I. INTRODUCTION

Pedagogical theory and practice considers laboratory experimentation an essential part of the educational process, particularly in sciences and engineering.

The increased capacity of the internet and wireless communication have changed the focus, from learning on the job or in the school, with the assistance of a person providing constant support, to a system where wireless mobile technologies can provide this assistance,not only in the workplace but also in any other location the learner chooses. Accordingly, laboratories are making their resources available via the Internet and are becoming part of important applications in the new and interdisciplinary field of Online Engineering [1]. The iLab Shared Architecture (ISA) is already established as a standardized implementation for online laboratories is becoming present in more and more locations, all over the world, as we speak.

The architectural developments and numerical increase of smart mobile devices, over the last years, offer the opportunity to make these online laboratories available to the end-user with much less device type or location limitations.

The Android operating system (OS) and its recent wide spread across many different mobile devices, coming from many different manufacturers, certainly contribute to this opportunity.

Also, while smart mobile devices could get access to the online laboratories through a general website, the authors strongly believe that a robust and OS specific client application is much more ergonomic, better performing and easier to use.

All these facts have motivated the authors to work on developing the first Android client application for an ISA enabled online laboratory.

A. *The iLab Shared Architecture*

The migration towards standardized solutions for delivering online labs is necessary to ensure software reusability and therefore ease online labs development and sharing.Such a standardized solution is the iLab Shared Architecture.ISA is a software architecture developed at the MIT (Massachusetts Institute of Technology) that facilitates a cross institution sharing and management of online labs.It offers a framework that establishes clear rules governing the communication between clients and their respective online lab servers by means of an API (Application programming Interface) based on web services SOAP calls.

From the perspective of online laboratory management, experiments fall into three broad categories [2]:

- *Batched experiments* are those in which the entire course of the experiment can be specified before the experiment begins.
- *Interactive experiments* are those in which the user monitors and can control one or more aspects of the experiment during its execution.
- *Sensor experiments* are those in which users monitor or analyze real-time data streams without influencing the phenomena being measured.

Several implementations of ISA enabled laboratories exist at this time. A few examples are presented in [3], [4], [5], [6] and [7].

The Android client application which is the subject of this paper connects to a laboratory that falls in the batched laboratory category.

B. *About the Android operating system*

The Android operating system (OS) has come a long way since Google acquired Android Inc. (2005) and the announcement ofthe Open Handset Alliance (2007) with the goal to build a better mobile phone.

Google has led a successful movement to turnthe closely guarded mobilephone market into one where mobile phone users can move betweencarriers easily and have unfettered access to applications and services.

Meanwhile, the Android OS has become present on a large number of other devices as well (tablets, media boxes, watches, televisions etc.) and this trend is still continuing. The reader can find, in [8], information regarding a few devices that run Android OS, including user input methods and device sensor information.

Currently, the most used versions of Android are 2.3 (Gingerbread) and 2.2 (Froyo), present on over 85% of the Android Market visitor devices [9].

Android is an operating system but also a software platform upon which applications are developed.Unlike some proprietary platforms that require developer registration fees, vetting, andexpensive compilers, there are no upfront costs to developing Android applications[10].

C. *Application context*

As mentioned above, this work proposes the implementation of a client application for the Android OS within the ISA framework and is focused onbatched experiments.

The client application was desired to allow users to carry out experiments from an Android device in an existing laboratorythat uses an analog programmable device. This laboratory iscalled “READ – Remote ASIC Design and Test” and was developed with an analog ASIC (Application-Specific Integrated Circuit). It allows users to remotely test and program the device over the internet as well as to create different circuits with a softwareapplication and upload them to the real-world device.

The laboratory was already accessible online by means of a Java applet integrated into ISA called READLabClient.

This applet was used as a reference for developing the new Android client application. However, it was clear from the beginning that the development process will come with its own challenges and an important one would be connecting to the Service Broker (since the current Service Broker version was not developed formobile application connectivity).

II. THE APPLICATION

A. *Development notes*

The Java Platform, Standard Edition Development Kit (JDK) version 6 formed the underlying development environment. The JDK is a development environment for building applications, applets, and components using the Java programming language.

The Android SDK is a prerequisite in order to be able to develop applications for Android with Java. It is important to note that the Android SDK that one initially downloads is only a starter package, not the full SDK.

Based on the available test hardware capabilities and to increase portability, the initial application is targeted at the 2.2 Android platform - API (Application Programming Interface) level 8.

To speed up development, Eclipse was the IDE (Integrated Development Environment) of choice, along with the ADT (Android Development Tools) plugin provided by Google. The ADT plugin was designed for the Eclipse IDE to extend its capabilities in order to facilitate setting up projects, ease GUI (Graphic User Interface) development and to allow easy debugging and exporting of Android applications. The Eclipse IDE version 3.7 (Indigo) has been used for the development process.

A part of the code of the already developed ReadLabClient Java applet was reused, due to similar functionality and target of the Android application (both are designed to interact with the same online laboratory).

The GraphView library was used for charting purposes.

Because the SOAP protocol is needed for accessing the web services provided by the Service Broker, for easy SOAP message composition and communication, an existing Java implementation of the protocol, developed at MIT, was slightly modified and used.

B. *Application functionality*

The application is divided in two important parts, each of them implemented as a separate Android activity:

- The login activity.
- The laboratory activity.

The login activity implements the user authentication on the Service Broker, in order to be able to proceed to the laboratory activity. It collects the user authentication data (username and password), sets additional parameters needed for the login process and submits the required datadata to the Service Broker. Based on this data, the Service Broker issues a response which will either confirm successful authentication or include error messages (such as incorrect login data). The login activity processes the response and launches the laboratory activity upon successful authentication confirmation.

The laboratory activity features a Graphic User Interfacewhich is an adaptation of the existing Java applet GUI for the Android OS. This GUI will be detailed in what follows. The laboratory activity core functionality consists of collecting the experiment user data, submitting it to the Service Broker (which, in turn, forwards it to the laboratory server), querying the execution status and processing the Service Broker responses (including the experiment results).

For an easier understanding of the application functionality, fig. 1 presents a very simplified application functionality flowchart. A more detailed flowchart will be included in an upcoming paper.

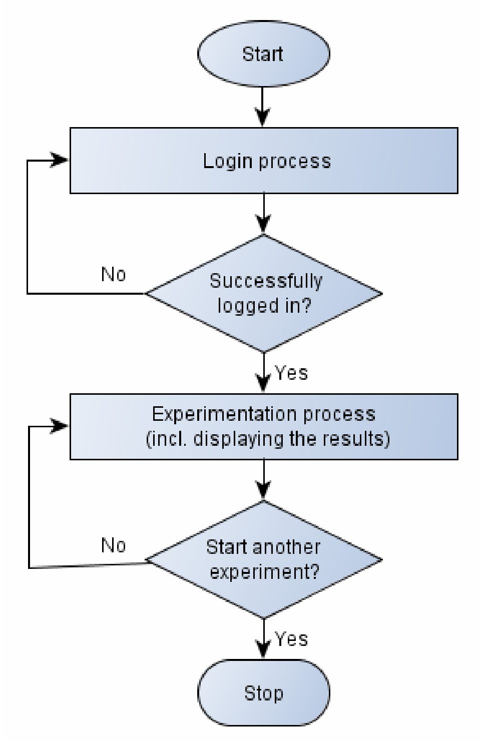
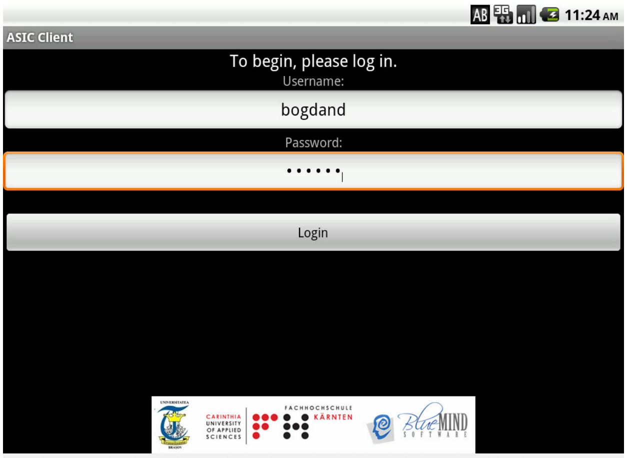

Start
Login process
No
Successfully logged in?
Yes
Experimentation process (incl. displaying the results)
No
Start another experiment?
Yes
Stop

Figure 1. Application functionality flowchart (part 2)



ASIC Client

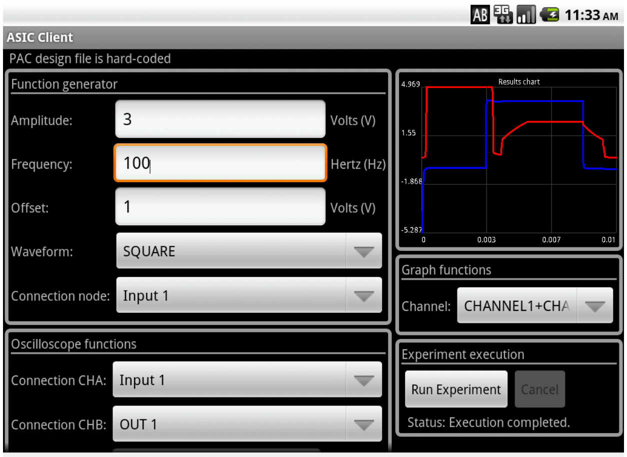
To begin, please log in.

Username:
bogdand

Password:
••••••

Login

Figure 2. The login activity (virtual device emulator screenshot)



ASIC Client

PAC design file is hard-coded

Function generator

Amplitude: 3 Volts (V)

Frequency: 100 Hertz (Hz)

Offset: 1 Volts (V)

Waveform: SQUARE

Connection node: Input 1

Oscilloscope functions

Connection CHA: Input 1

Connection CHB: OUT 1

Graph functions

Channel: CHANNEL1+CHA

Experiment execution

Run Experiment Cancel

Status: Execution completed.

Figure 3. The laboratory activity (virtual device emulator screenshot)

The application benefits of a multithreaded implementation for performance increase and smooth functioning.

Fig. 2 presents a screenshot of the login activity graphic user interface. The user only has to input his username and password. Upon pressing the Log In button, the activity connects to the Service Broker and handles the authentication process.

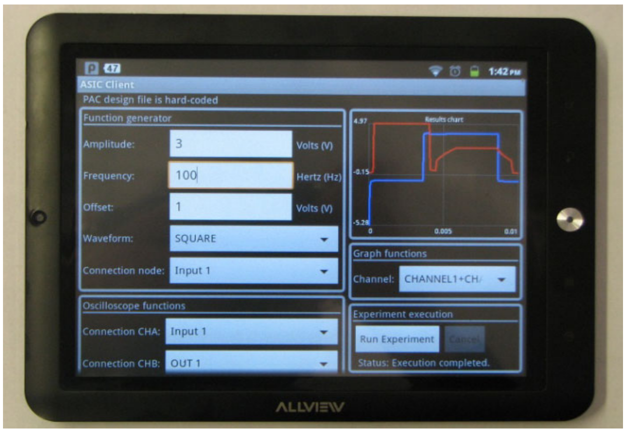
Because this is the first Android application that connects to an ISA integrated laboratory, at the time of development there was no standard authentication method designed specifically for external mobile clients. The login procedure simulates standard browser login interaction and was determined by reverse engineering online client implementations. It was the hardest part of the development process and may be detailed in an upcoming paper. Work is currently being done in order to develop a much easier standard authentication protocol for 3rd party clients (desktop and mobile clients).

Fig. 3 presents a screenshot of the laboratory activity graphic user interface. The GUI is divided in two main parts: experiment parameters on the left side of the screen and experiment results and controls on the right side of the screen. The experiment parameters are specified by the user prior to sending them to the service broker and will determine the whole course of the experiment execution. The user cannot change these parameters during experiment execution.

To start, the user has to input experiment specific data, on the left side of the screen, divided in 3 categories:

- PAC design file (hardcoded at this time, but will be interactive in the near future).
- Function generator parameters.
- Oscilloscope parameters.

After inputting the experiment data, the user presses the "Run Experiment" button to submit the experiment data. The application will check the experiment execution status (by querying the Service Broker which, in turn, queries the laboratory server) at predefined intervals, until the experiment execution is completed. The user may cancel the experiment execution at any time, in this period. When the experiment results are ready, the application gets, processes and displays them in an intuitive format (the chart situated in the top-right side of the GUI).



ASIC Client

PAC design file is hard-coded

Function generator

Amplitude: 3 Volts (V)

Frequency: 100 Hertz (Hz)

Offset: 1 Volts (V)

Waveform: SQUARE

Connection node: Input 1

Oscilloscope functions

Connection CHA: Input 1

Connection CHB: OUT 1

Graph functions

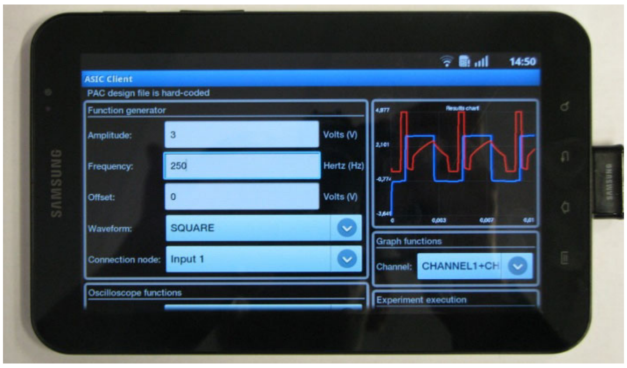
Channel: CHANNEL1+CH/

Experiment execution

Run Experiment Cancel

Status: Execution completed.

Figure 4. The application deployed on an AllView Alldro2 tablet (with visible experiment results)



ASIC Client

PAC design file is hard-coded

Function generator

Amplitude: 3 Volts (V)

Frequency: 250 Hertz (Hz)

Offset: 0 Volts (V)

Waveform: SQUARE

Connection node: Input 1

Oscilloscope functions

Graph functions

Channel: CHANNEL1+CH

Experiment execution

Figure 5. The application deployed on a Samsung Galaxy Tab tablet (also with visible experiment results)

C. *Deployment and tests*

The application was not only tested in the virtual device emulator but also on real-life tablets: an Alldro 2 tablet and an Alldro Speed tablet, both manufactured by AllView (with Android 2.3 Gingerbread), and a Galaxy Tab manufactured by Samsung (with Android 2.2).

Fig. 4 presents the application deployed on the AllView Alldro2 and fig. 5 shows it deployed on the Samsung Galaxy Tab. All pictures were taken after successful experiment executions.

Although the current GUI was designed for devices with larger screens, a mobile phone device (AllView P1, with Android 2.2 Froyo) was used for tests as well, with good functional results.

III. CONCLUSIONS

This paper presented the very first Android client application for ISA laboratories. As demonstrated, this version of the application manages to authenticate with the Service Broker and allows its users to communicate and conduct remote experiments with one of the existing batched laboratories. The application was successfully tested on a few real-world mobile devices.

For now, the application had to use specialized and complex methods in order to authenticate with the current version of the Service Broker. Using the experience gained during the application design and development process, a new standardized web service authentication method will be developed. Future versions of the Service Broker will include this method and, thus, will allow easy authentication from desktop and mobile applications.

The application can be used in the pedagogical process at CUAS (Carinthia University of Applied Sciences) in conjunction with the Remote ASIC Design and Test laboratory. The experienced gained will prove useful for designing and developing other remote client applications for the Android operating system.

IV. FUTURE RESEARCH AND DEVELOPMENT

Although this first version of the application is perfectly functional and optimized for best performance, more user input error handling is required to achieve the desired software system robustness. Also, the best way to insert user created PAC design files is being discussed.

The application will implement the new web service authentication method for the future versions of the Service Broker, when the method will be ready.

From the general viewpoint of remote Android client applications, future Android applications will target interactive laboratories and other remote monitoring and control systems, according to the long term research goals (incl. the tele-manufacturing systems proposed in [11]).

ACKNOWLEDGMENT

The authors would like to thank AllView for hardware support and Bluemind Software for application development expertise.

REFERENCES

[1] M. E. Auer, A. Edwards, D. G. Zutin,“Online laboratories in interactive mobile learning environments”. In *Work-Based Mobile Learning – Concepts and Cases,* N. Pachler, C. Pimmer, J. Seipold (Eds), Peter Lang AG, 2011, ISBN 978-3-03911-982-0, pp. 219-248.

[2] J. Harward et al., “iLab: A Scalable Architecture for Sharing Online Experiments”, proc. of the International Conference on Engineering Education (ICEE) 2004, Gainesville, FL, USA, October 16–21, 2004, pp. 641-645.

[3] D. G. Zutin, “Networking Online Labs within the ISA Framework”, *iJOE*, Vol. 5, Issue 4, 2009, pp. 20-23.

[4] H. Soumare et al.,“AVersatile Internet-Accessible Electronics Workbench with Troubleshooting Capabilities”, *iJOE*,Vol. 5, Special Issue REV 2009, 2009, pp.72-80. http://dx.doi.org/10.3991/ijoe.v5i4.1084

[5] D. G. Zutin, M. E. Auer, I. Gustavsson, „A VISIR lab server for the iLab Shared Architecture “,*iJOE*, Vol. 7, Special Issue EDUCON 2011, 2011, pp. 14-17. http://dx.doi.org/10.3991/ijoe.v7iS1.1754

[6] D. Ursutiu et al., “Multifunction iLab Implemented Laboratory”, proc. of the IEEE Global Engineering Education Conference (EDUCON) 2010, Amman, Jordan, April 4-6, 2010, pp. 185-190. http://dx.doi.org/10.1109/EDUCON.2011.5773135

[7] D. Pop at al., “An Online Lab to support a Master Program in Remote Engineering”, proc. of the 41st ASEE/IEEE Frontiers in Education Conference, GOLC2, Rapid City, SD, October 12 - 15, 2011, pp. 1-6.

[8] J. Steele, N. To, *The Android Developer’s Cookbook. Building Applications with the Android SDK*, Pearson Education, Inc., 2011, ISBN-13: 978-0-321-74123-3.

[9] Google, “Platform Versions | Android Developers”, http://developer.android.com/resources/dashboard/platform-versions.html. Accessed on: 10.01.2012.

[10] S.Conder, L.Darcey, *Android™ Wireless Application Development*, Pearson Education, Inc., 2011, ISBN 978-0-321-74301-5.

[11] B. Deaky, N.B. Lupulescu, M. Dragoi, Objectives of Telemanufacturing Research at AMTS, UniTBv. Initial Achievements, *AJME*, Vol. 8, Issue 3/2010, pp. 12-17.

AUTHORS

B. Deaky is an intern post-doctoral researcher at Carinthia University of Applied Sciences, Villach, Kaernten A-9500 Austria, on leave from the AMTS (D05) research department of the Transilvania University of Brasov, Romania. He is in chargefor the local research in the tele-manufacturing field, in Romania. (e-mail: bogdan_deaky@xu.unitbv.ro).

D. G. Zutin is an Assistant Professor and senior researcher at the Carinthia University of Applied Sciences, Villach, Kaernten A-9500 Austria (e-mail: dgzutin@IEEE.org).

P. H.Bailey is with the Center for Educational Computing Initiatives, Massachusetts Institute of Technology, Cambridge, MA, USA (e-mail: pbailey@mit.edu)

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the project number POSDRU/89/1.5/S/59323. Submitted, 17 January 2012. Published as resubmitted by the authors on 9 February 2012.