**Language for User Created Services on Android Platform**

Chia-Hao, Shih, Omer Boyaci and Henning Schulzrinne

Internet Real time Laboratory

Columbia University, NY, 10027, USA

cs2805@columbia.edu, {boyaci, hgs}@cs.columbia.edu

**Abstract**

With more and more communication application used today, sensors and actuators are also becoming popular. LUCS, a new scripting language and support infrastructure which will enable users to create services for controlling their communication flow integrating different sensors and services is developed now. This report discusses our diagnostics visualization system for LUCS on android platform.

1. **Introduction**

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The [Android SDK](http://developer.android.com/sdk/) provides the tools and APIs necessary to begin developing applications that run on Android-powered devices. The Google APIs add-on extends Android SDK to give applications access to Google libraries such as Maps. Our goal is to use the Google API to keep track of user’s current location and transmit it to presence server via Session Initiation Protocol.

1. **Architecture**

2.1 Project Architecture

The system’s architecture can be divided into two main parts. In receiving part, the android phone receives the location message via location provider by using Google API. In transmission part, the android phone sends the location message to presence server by using Jain-Sip library. This application is composed of these two parts. Figure 1 shows the architecture of this project. Android platform receives the location message including latitude and longitude and transmit the message to server in PIDF(Presence Information Data Format) format with SIP protocol.



Figure 1:Archtecture of Project

* 1. Receiving Part

This part is implemented in the InstantLocation.java. This part has been finished. In this application, user’s location will update periodically by setting

locationManager.requestLocationUpdates(provider, long minTime, float minDistance, locationListener);

The frequency of notification may be controlled using the minTime and minDistance parameters. If minTime is greater than 0, the LocationManager could potentially rest for minTime milliseconds between location updates to conserve power. If minDistance is greater than 0, a location will only be broadcasted if the device moves by minDistance meters. I set minTime = 2000 and minDistance = 10 in this application. Figure 2 shows all available providers now and the current position in PIDF format.



Figure 2: Instant Location Message

* 1. Transmission Part

This part is implemented in the SipLayer.java. This part has not finished yet. In order to implement the Sip protocol with JAIN-SIP protocol.

Application Responsibility

• Application registers an implementation of the SipListener interface to interact with the SIP Stack.

• Application must register with the SipProvider for all messaging capabilities with the stack.

– Application requests transactions for stateful messaging.

– Application sends stateless messages.

– Access stack objects.

• Application receives messages from the stack as Events via the SipListener interface.

Figure 3 shows the JAIN-SIP object architecture.



Figure 3: JAIN-SIP Object Architecture

**5.     Experiment and Results**

 I have fully tested the receiving part which is implemented in InstantLocation.java to check whether the user’s location is the same as the Google Map. The accuracy is 99%, and the data is as follows:



**6. Future Work**

The future work for this project is to finish the transmission part which is implemented in SipLayer.java including sending the location message to presence server and receiving ACK from server.

**7.**     **Acknowledgement**

I would like to thank Victoria Beltran for providing us with valuable guidance at every stage of the project.

8.     **Reference**

[1] Google Maps API, <http://code.google.com/intl/en/apis/maps/>

[2]JAIN-SIP API 1.2, <http://hudson.jboss.org/hudson/job/jain-sip/lastSuccessfulBuild/artifact/javadoc/index.html>

[3] Reto Meier, Professional Android Application Development, ISBN: 978-0-470-34471-2

[4]National Institute and Standards and Technology, <http://snad.ncsl.nist.gov/>

[5]Java.Net, <http://www.java.net/>

[6] Havard Sataoen, Torbjorn Strom, “Location Based Services in mobile Java applications A comparative study of Java Micro Edition and Android”, Buskerud University College Kongsberg, Norway, 2008

**9.**     **Appendix: Program Documentation**

**8.1**   **Executables**

InstantLocation.apk: android application installed in android phone

**8.2**   **src**

InstantLocation.java: receive the location message from location provider.

SipLayer.java: transmit the location message to the presence server.

MessageProcessor.java: an interface defines all the method which is used to process the message and error in the SipLayer.java.

**8.3** **gen**

R.java: auto-generation files generated in terms of all xml files and icons used in this application.

**8.4 res**

 Layout: main.xml: define the layout of this application.

 Values: stinng.xml: define the strings used in the layout of this application.

 AndroidManifest.xml: list all the function this application provided including GPS, Internet.

**8.5Assets**

Jain-sip-api-1.2: version1.2 JAIN-SIP library