Improving Rural Early Warning Systems through the Integration of OpenBTS and JAIN SLEE[*]

Julián Eduardo Plazas[**]
Juan Sebastián Rojas[***]
Juan Carlos Corrales[****]

Abstract

Nowadays exists a trend that combines the features of Web 2.0 services and telecommunications services known as Telco 2.0. These converged services have been successfully implemented in early warning systems providing improved agility and flexibility in service delivery. However the deployment of converged services in rural zones of developing countries presents several constraints which do not allow to provide this kind of services, as the unavailability of a Next Generation Network (NGN), absence of advanced technology and lack of investment resources. This paper proposes a JAIN SLEE and OpenBTS integration architecture for early warning systems in rural zones. The implemented prototype is evaluated with a specific case study involving the deployment of Telco 2.0 warnings in Colombian coffee plantations which may be affected by coffee rust, one of the most threatening diseases in coffee production.

Key words: JAIN SLEE, OpenBTS, integration, early warning system, converged services

[**] Ph.D student in telematics engineering, B.S. in electronics and telecommunications engineering, Grupo de Ingeniería Telemática, Universidad del Cauca, Popayán, Colombia, Carrera 6 B # 28 BN 16, Popayán–Cauca, Tel: (+57-2) 8367077, E-mail: jeplazas@unicauca.edu.co

[***] Ph.D student in telematics engineering, B.S. in electronics and telecommunications engineering, Grupo de Ingeniería Telemática, Universidad del Cauca, Popayán, Colombia, Carrera 17 # 19N-166B, Popayán–Cauca, Tel: (+57-2) 8230443, E-mail: jsrojas@unicauca.edu.co

[****] Ph. D in Computer Science, full time professor at Universidad del Cauca, Coordinator of Grupo de Ingeniería Telemática, Popayán, Colombia, Edificio Ingenierías, Tulcán, Oficina 403, Tel: (+57-2) 8209800, ext. 2129, E-mail: jcorral@unicauca.edu.co

Mejorando los sistemas rurales de alertas tempranas a través de la integración de OpenBTS y JAIN SLEE

Resumen

Actualmente existe una tendencia que combina las características de los servicios Web 2.0 y los servicios de telecomunicaciones, conocida como Telco 2.0. Estos servicios convergentes se han aplicado exitosamente en sistemas de alertas tempranas, proporcionando mayor agilidad y flexibilidad en la prestación de servicios. Sin embargo, existen varias limitantes que no permiten el despliegue de servicios convergentes en las zonas rurales de países en vía de desarrollo, como la falta de disponibilidad de una NGN (Next Generation Network), la ausencia de tecnología avanzada y la falta de recursos para inversión. Este artículo propone una arquitectura de integración entre JAIN SLEE y OpenBTS para sistemas rurales de alertas tempranas. Se evalúa el prototipo implementado con un caso de estudio específico al enviar advertencias Telco 2.0 a los cafeteros colombianos cuyas plantaciones puedan verse afectadas por la roya, una de las enfermedades más peligrosas para la producción de café.

Palabras clave: JAIN SLEE, OpenBTS, integración, sistema de alertas tempranas, servicios convergentes.

INTRODUCTION

Currently, there is a trend in which the policies and efforts implemented by emergency managers are being focused on decreasing their response time before emergency situations. With this objective in mind, different methods have been proposed, among which prevention and early detection of risk situations are the most important. This new paradigm seeks to leverage the most popular technologies nowadays, such as telecommunications and Web 2.0 services that offer an important advantage in the dissemination of information due to the large number of people who use such technologies, although not being centered in emergency response [1]. Within the Web 2.0 services, social networks like Facebook and Twitter can be very useful for emergency management by sharing enormous amounts of multimedia information, like videos and pictures, to hundreds of thousands of users in a matter of seconds [1]. Otherwise, in the telecommunications services, mobile services might be the most useful for emergency management since they enable the possibility to communicate the status of risky situations to large amounts of people within a coverage area [2].

Although the differences between Web 2.0 services and telecommunications services, in the ICT (Information and Communication Technologies) a new model known as Telco 2.0 has been defined, proposing the integration of concepts, technologies and services of Web 2.0 with traditional telecommunications' features. Therefore, the services inside the Telco 2.0 model are known as converged services [3]. The converged environments represent an advantage for early warning systems because they offer improved agility and flexibility in service delivery, *i.e.,* adding Web 2.0 functionalities to traditional telecommunications services not only provides a more efficient dissemination of warnings, it also allows to detect and manage them without increasing the system's complexity.

However, rural zones of developing countries present significant difficulties when it comes to provide converged services features, since a Next Generation Network (NGN) is needed for their deployment, and this kind of networks are specially oriented to cities with large amounts of population. Furthermore, the lack of advanced technology and investment resources caused by the absence of industry enterprises which do not find profitable outcomes when investing in these areas. This makes the implementation of all kinds of projects a much more difficult task. Notwithstanding, there are several low cost alternatives which would allow to deploy converged services in rural zones, like femtocells, OpenBTS tools and the JAIN SLEE specification.

Considering the previous statements, this paper presents a proposal focused on the integration of the JAIN SLEE specification –a standard and robust environment for

the creation and execution of converged services–, and the OpenBTS architecture supported by Range Networks –which allows the establishment of a GSM network based on open source software and low costs hardware–, in order to develop an early warning system capable of detecting and triggering alerts through three specific services (voice call, SMS and email) implemented on the JAIN SLEE environment.

This paper is arranged as follows. The next section presents a conceptual base of the different technologies related with this work. Section 2 describes the proposed integration architecture. Section 3 presents a detailed description of the experimental prototype and a case study through which the prototype is evaluated. Finally, section 4 presents the conclusions obtained along with possible future works are exposed.

1. BACKGROUND

Before describing the architecture and implementation proposed by this paper, it is necessary to define and comprehend some of the most important concepts, components and technologies surrounding the JAIN SLEE specification, the OpenBTS tools, and the early warning systems, which will be briefly described as follows.

1.1 JAIN SLEE Specification

The JAIN SLEE specification is composed by two concepts which must be analyzed independently. Firstly, the Service Logic Execution Environment (SLEE) which satisfies the requirements needed to run a telecommunications service or application, fulfilling the intrinsic parameters of their nature, such as low latency and high performance. These capabilities also allows the SLEE to support the execution and deployment of converged services, by combining web and telecommunications attributes, creating more complex and dynamic services. The second concept is Java API for Integrated Networks (JAIN) standard. This standard defines a set of Java API aimed at developing telecommunication services, including aspects ranging from signaling and call control, to the availability of high-level programming interfaces, ensuring that services can be deployed seamlessly, regardless of the network technologies. Therefore, these API must provide three principles: service portability, network convergence and access to network resources [4-5].

Considering the analysis of these concepts, it is not difficult to understand how the JAIN SLEE [6] specification defines a standard execution environment for service logic, and specifies the manner in which robust converged services can be built, managed and executed, decreasing time-to-market by providing a standard programming model aimed at the community of Java developers [4].

1.2 OpenBTS

Open base transceiver station (OpenBTS) is an open source project developed by Range Networks, whose main goal is to implement an infrastructure capable of establishing a functional mobile network connection, apart from a traditional operator. Thus, besides providing significant cost savings, it allows to customize a second generation radio frequency access. This application consists of a software based GSM access point which allows its IP PBX (private branch exchange) to detect GSM mobile phones as if they were SIP terminals in a VoIP (voice over IP) network [7], managing to replace traditional network stations with easy installation and lower cost ones, where each user of this new network may establish intra / inter network connections with other users through Internet. It is important to notice that mobile phones continue to operate under the GSM technology but through an OpenBTS system, not a traditional base transceiver station [8].

The most important software and hardware components for the installation of this kind of stations are: the universal software radio peripheral (USRP), used to present a GSM interface using the GNU radio framework; the Asterisk open source PBX, used to manage the call control; OpenBTS application; and a computer with Linux operating system [9].

1.3 Early Warning Systems

In disaster management, early warnings are important to reduce the harmful impacts of a disaster, empowering individuals and communities threatened by hazards to act in sufficient time and in an appropriate manner to reduce the possibility of personal injury, loss of life and damage to property and the environment. Therefore, an early warning system needs to actively involve the communities at risk, facilitate public education and awareness of risks, effectively disseminate messages and warnings, and ensure there is constant state of preparedness [10].

A complete and effective early warning system comprises four key elements: 1) risk knowledge, a component which allows the system to assess the actual risk of a situation due to its emergency related collected information; 2) monitoring & warning service, a full time operating component capable of performing real time analysis on the received information in order to generate accurate warnings in a timely fashion; 3) dissemination & communication. This component handles all the procedures needed to deliver the warnings to the final user through multiple channels or services, ensuring a more complete dissemination, avoiding failures and reinforcing the warning message; and 4) response capability. This component represents the knowledge and response mechanisms the population has, *i.e.,* how people must react before a danger-

ous situation and which tools they have to face the phenomenon (escape routes, first aid kits, etc.) [10].

It is worth mentioning that this paper proposes the implementation of one of the four key elements of an early warning system, the dissemination & communication component through the services implemented inside the converged environment along with OpenBTS.

2. ARCHITECTURE

Figure 1 shows the integration architecture based on the example found in [4], which is composed by:

- **Converged control module:** this module handles the control and signaling operations needed by the converged services, it is composed by the service container and the service control units. The service container manages all the logic and resources related with the service behavior and communication with other components. On the other hand, a service control unit is a modular component inside the service container that defines how and when a service is executed.

- **Telco control module:** this module handles the control and signaling operations needed to run the telecommunications services. It is composed by service control units which can support one or more traditional Telco services, like voice call or sms.
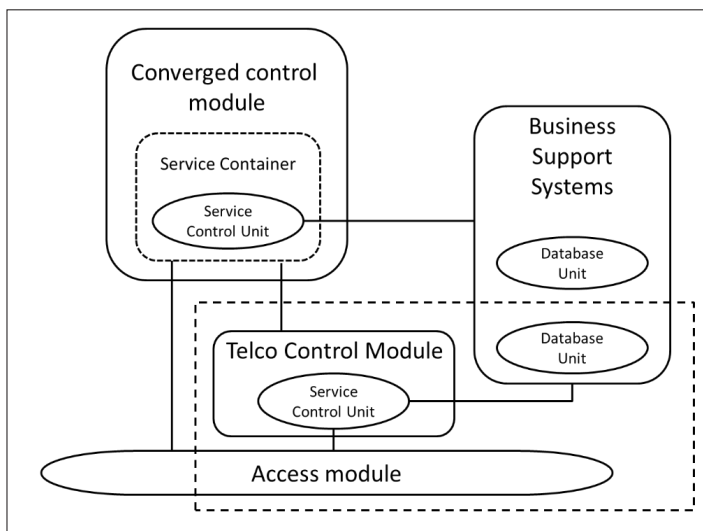

Figure 1. Integration architecture
Source: authors

- • **Access module:** this module refers to all the transport nodes within the integration architecture needed by users to access the services' functionalities.

- • **Business support systems:** the support system represents all the features necessary to store the subscribers' information and configuration parameters of the network.

3. CASE STUDY AND EXPERIMENTAL PROTOTYPE

Coffee production is the main agricultural activity in Colombia, where more than 350,000 families depend on coffee harvest for their sole income. Diseases, pests and weather changes cause a big impact in coffee production, triggering a significant reduction in these families' quality of life [11]. The most important disease affecting Colombian coffee production is coffee rust, currently infecting more than 50 percent of the country's coffee, even though varieties of coffee that could endure this disease have been developed through improvement with genes. Studies on coffee rust have concluded that the spores carrying the infection are spread by climatic elements such as wind and rainfall. Wind is the vector for long distance spore transport, whilst precipitation droplets are responsible for vertical propagation from infected leaves or soil [11].

Deploying warnings in rural areas of developing countries, like Colombian coffee plantations, is not an easy task, due to their limited telecommunications infrastructure, hence the use of open source technologies like OpenBTS and the JAIN SLEE specification offers the possibility of developing an early warning system capable of preventing different types of difficulties like coffee rust by watching over a set of climatic parameters (like temperature, humidity, precipitation) and once a risky situation is detected, proceeding to alert the farmers and the people involved with the plantation through converged services.

Considering the previously described case study, we developed an experimental prototype following the architecture observed in section 3. This prototype is designed to deploy warnings aimed to eight people (one person per family within the coverage area) through three services implemented within the JAIN SLEE specification: voice call, SMS and email. Table 1 presents the hardware features and figure 2 illustrates the deployment design of the experimental prototype which is explained as follows.

Table 1. Hardware features of the experimental prototype

| Component	Quantity and Model	Description
USRP	1 - N210	This component allows the establishment of a GSM network supported by the OpenBTS software.
GSM phones	3 - LG (Smartphone) 3 - Samsung (Smartphone) 4 - Nokia Lumia (Smartphone) 1 - Sony Xperia V (Smartphone) 1 - Motorola Moto G (Smartphone) 1 - Motorola W375 (Cellphone) 2 - Nokia X3 (Cellphone)	These phones connect to the GSM network and allow the testing of the Telco services.
Computer	Dell-PowerEdge T100	This component runs all the OpenBTS software and the JAIN SLEE environment.

Source: authors
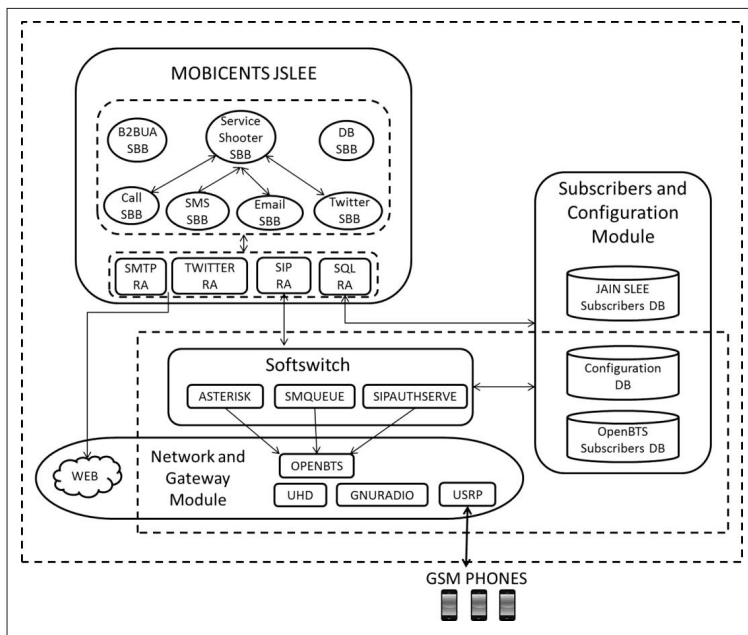
Figure 2. Deployment design of the experimental prototype.
Source: authors

Mobicents JSLEE: this module refers to the converged control module presented in section 3 composed by the service container. The service container is divided in the Service Logic Container (contains the service building blocks) and the Service Adaptation Container (contains the resource adaptors). The SBB needed for this prototype are:

- DB SBB: the Database Service Building Block communicates with the subscribers' databases through the SQL RA and allows other SBB to access all the information stored in them.

- B2BUA SBB: the Back to Back User Agent SBB operates like a SIP event router between the softswitch and the other SBB, by mediating all the signaling amid them and dividing each communication channel into two call legs.

- Call SBB: this SBB handles the SIP signaling needed to initiate a voice call by sending an invite message to a predefined callee and establishing a session between both end points.

- SMS SBB: the short message service SBB creates and sends a SIP instant message to a predefined user without establishing a session. It is worth mentioning that both the call SBB and the SMS SBB are supported by the B2BUA SBB.

- Email SBB: this SBB sets all the headers and properties needed to send an email to a predefined user via Simple Mail Transfer Protocol (SMTP) from a Gmail account.

- Twitter SBB: this SBB is capable of posting a warning message in an official Twitter account for the project through the Twitter RA.

- Service Shooter SBB: this SBB is intended to sequentially trigger the SMS SBB, the Call SBB and the email SBB with the information of all the users provided by the DB SBB. For the time being, this SBB is manually activated to run the performance tests.

The RA needed for this prototype are:

- SIP RA: this resource adaptor provides the SLEE with the capabilities to handle all SIP events.

- SQL RA: this resource adaptor provides the SLEE with the capabilities to connect with SQL databases, mainly SQLite.

- Twitter RA: this resource adaptor provides the SLEE with the capabilities to access Twitter functionalities through an API key.

- SMTP RA: this resource adaptor provides the SLEE with the capabilities to open a SMTP session with a preset email server.

Softswitch: this module refers to the Telco control module presented in section 3. It is the central device in the OpenBTS network which connects the telephone calls and other services from one phone line to another through three main components:

- **Asterisk:** this component handles the commutation of all voice calls by leveraging the SIP protocol.

- **Smqueue:** this component stores and forwards all text messaging by using the message SIP request method.

- **Sipautserve:** this component handles all SIP registration and authorization requests, it is also used to process location update requests from OpenBTS and perform corresponding updates in the subscriber registry database.

Network and gateway module: this module refers to all the transport nodes from the OpenBTS infrastructure and the OpenBTS software features which handle the communication between SIP and GSM protocols, including user register, establishment of voice calls and instant messaging requested by network users. It consists of four components:

- **OpenBTS:** this component handles all incoming requests from GSM endpoints, translates them into SIP, and communicates with any of the previous components according to the GSM request received (call, SMS, registration).

- **USRP:** the universal software radio peripheral is intended to be a comparatively inexpensive hardware platform for software radio. It connects to a host computer through a high-speed link, which the host-based software uses to control the USRP hardware and transmit and receive data.

- **UHD:** the USRP hardware driver is the hardware driver for the USRP device. The goal of the UHD software is to provide a host driver and API for the communication and control of the USRP so the OpenBTS software can perform the translation between GSM and SIP protocols.

- **GNURadio:** this is an open source software development toolkit that provides signal processing blocks to implement software radios, used with readily-available low-cost external RF hardware to create software-defined radios, hence GNURadio enables UHD drivers to establish a communication with the USRP.

Subscribers and configuration module: this module refers to the support systems already explained in section 3. Within them there are three components:

- JAIN SLEE subscribers DB: this database contains all the information related to the users, needed within the JAIN SLEE services.

- **OpenBTS subscribers DB:** this database contains all the information related to the users, needed within the OpenBTS architecture.

- • **Configuration DB:** this database contains the entire configuration data needed within the OpenBTS architecture.

In order to evaluate the performance of the experimental prototype, twelve tests were conducted observing the execution time (how much time the service needs to complete all its processing for each user) and the success rate (how many users successfully received the service). Considering that this prototype is aimed for eight users, three tests were performed considering three different alert levels: level 1 for low risk alerts, email and twitter post; level 2 for medium risk alerts, email, twitter post and SMS; level 3 for high risk alerts, email, twitter post, SMS and voice call. Each test was repeated ten times and is divided according to the number of users (1 user, 5 users, 10 users and 15 users). Figure 3 illustrates the results obtained in the tests performed. The orange line stands for the success rate in terms of percentage and the green bars stand for the execution time measured in seconds and is divided according to the user's number in each test performed.

Figure 3. Experimental prototype test results.
Source: authors

Three graphics can be observed in Figure 3, considering three different alert levels. Level 1 alert test results graphic shows a decreasing execution time with a little rise on the 15 users test due to the processing of the email SBB, which sends emails to groups

of up to 10 users. This graphic also shows an ideal success rate with a 100 % score. Level 2 alert test results graphic shows a decrease in the execution time when the users number increases, but it does not increase on the 15 users test, this happens because the SMS service is very light in terms of processing, hence accessing the database is a critical stage in the execution time. The success rate also has a decreasing trend caused by difficulties with the USRP, which could not establish a communication with the phones within the GSM network, showing a user unreachable exception. Level 3 alert test results graphic has the same behavior in the execution time than level 1 alert test results graphic. This happens because the voice call service has an even execution time on every test, furthermore since the execution time of the voice call service is bigger than the SMS, the SMS does not have a relevant impact in the obtained results. The success rate has a little decrease caused by the aforementioned USRP difficulties. Notwithstanding the success rate drops, it does not fall under the 85 % on any of the three alert levels, being good enough for a rural early warning system.

CONCLUSIONS AND FUTURE WORK

This paper presented an architecture aimed for the integration of the JAIN SLEE specification with the OpenBTS tools. In order to evaluate our proposal, an experimental prototype of an early warning system dissemination & communication component with four services (voice call, SMS, email and Twitter) was implemented and tested in terms of execution time and success rate. From the obtained results observed in figure 3 and analyzed in section 3, it is possible to conclude that a successful integration between the JAIN SLEE specification and the OpenBTS tools was achieved, obtaining satisfactory results in both the execution time and the success rate. Such results could be improved by enhancing the hardware used for GSM connection, since the main fail reason (user unreachable) occurs when the USRP cannot find a phone.

Integration of JAIN SLEE specification and the OpenBTS tools is a feasible alternative for the implementation of converged services in rural early warning systems.

As future works, we propose, in the first place, the addition of monitoring functionalities by integrating a complex event processing facility within the proposed architecture, which will allow the automatic triggering of the alerts by analyzing specific parameters received in a data stream provided by a sensor network. Secondly, a performance comparison between our proposal, developed with open source and low cost tools, and a similar project, implemented with commercial software such as OpenCloud's RHINO [12], in order to validate the technical and financial viability for an actual implementation of a rural early warning system.

ACKNOWLEDGEMENTS

REFERENCES

[1] A. Crowe, *Disasters 2.0: The Application of Social Media Systems for Modern Emergency Management*. United States: CRC Press, 2012.

[2] J. Vázquez, I. Santa & J. Restrepo, "Prototipo de una estación celular portátil para atención de emergencias". [Online]. Available: http://docplayer.es/2368053-Prototipo-de-una-estacion-celular-portatil-para-atencion-de-emergencias.html

[3] J. Rojas, L. Ordóñez-Ante & J. Carlos, "Web and Telco Service Integration: A Dynamic and Adaptable Approach", *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 5, 2014.

[4] J. C. Corrales, *Tecnologías para el desarrollo de servicios convergentes,* Cauca: Universidad del Cauca, 2012.

[5] Oracle. JAIN-Reference [Online]. Available: http://www.oracle.com/technetwork/java/index-137964.html

[6] JSR-000240 JAIN SLEE 1.1 - Final Release [Online]. Available: http://download.oracle.com/otndocs/jcp/jain_slee-1_1-final-oth-JSpec/

[7] J. R. E. Schooler, "SIP: Session Initiation Protocol", *RFC3261* [Online], June 2002. Available: https://www.ietf.org/rfc/rfc3261.txt

[8] *Wikipedia, the free encyclopedia* (2015, Feb. 3). OpenBTS [Online]. Available: https://en.wikipedia.org/wiki/OpenBTS

[9] OpenBTS.org. OpenBTS Public Release [Online]. Available: http://openbts.org/w/index.php/Main_Page

[10] *Developing Early Warning Systems: A Checklist, EWC III Third International Conference on Early Warning* [Online], Bonn, Germany, Mar. 27-29, 2006. Available: http://www.unisdr.org/2006/ppew/info-resources/ewc3/checklist/English.pdf

[11] D. C. Corrales, A. Ledezma, A. J. Peña, J. Hoyos, A. Figueroa & J. C. Corrales, "A new dataset for coffee rust detection in Colombian crops base on classifiers", *Sist. Telemática*, vol. 12, no. 29, pp. 9-23, jun. 2014.

[12] Open Cloud. Rhino Documentation [Online]. Available: https://developer.opencloud.com/devportal/display/RD2v2/Rhino+Documentation