Detection of Risky Situations and Abnormal Events in Early Warning Platforms Based On Complex Event Processing

Julián Eduardo Plazas Pemberthy
Juan Sebastián Rojas Meléndez

Trabajo de grado en ingeniería electrónica y telecomunicaciones

Director:
Juan Carlos Corrales Muñoz
PhD. En Ciencias de la Computación

Universidad Del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática

Popayán, Julio de 2015

Julián Eduardo Plazas Pemberthy
Juan Sebastián Rojas Meléndez

Detection of Risky Situations and Abnormal Events in Early Warning Platforms Based On Complex Event Processing

Trabajo de grado presentado en la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de

Ingeniero en:
Electrónica y Telecomunicaciones

Director:
Juan Carlos Corrales Muñoz
PhD. en Ciencias de la Computación

Popayán
2015

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones

TRABAJO DE GRADO – MODALIDAD TRABAJO DE INVESTIGACIÓN

FORMATO G: ACTA DE PRIMERA (X) SEGUNDA ( ) SUSTENTACIÓN

LOS JURADOS DEL TRABAJO DE GRADO TITULADO:
Detection of Risky Situations and Abnormal Events in Early Warnings Platforms Based on Complex Event Processing

HACEN CONSTAR:

Que siendo las 3:00 pm del día 03 del mes de septiembre de 2015 realizaron la Sustentación del Trabajo de Grado, (el) (los) señores (as) Julian Eduardo Plazas y Juan Sebastian Rojas

Bajo la dirección del Ingeniero: Juan Carlos Corrales

OBTENIENDO EL CONCEPTO DE:

APROBADO (X) APROBADO CON OBSERVACIONES ( ) APLAZADO ( )

NO APROBADO ( )

Para constancia se firma en Popayán, a los 03 días del mes de septiembre de 2015

JURADO COORDINADOR: Javier Alexander Hurtado
(Nombre)

(Firma)

JURADO: Ricardo Salazar
(Nombre)

(Firma)

Números de Recibo de Tesorería (Derechos de Sustentación - Acuerdo 41/89):
Juan Sebastian Rojas 23552973
Julian Eduardo Plazas 23552972

Dedicado a todas esas personas,
que con su apoyo,
hicieron posible la culminación de este trabajo.

Acknowledgements

The authors would like to thank University of Cauca and Phd. Juan Carlos Corrales Muñoz for their guidance and advisement in the development of this undergraduate thesis. The authors also are thankful with Mag. Eduardo Rojas Pineda and Mag. David Camilo Corrales Muñoz for all their support and collaboration.

Structured abstract

The early warning systems are currently being implemented inside converged environments, however an improved alert dissemination is not enough for this kind of systems due to the necessity of processing and analysing asynchronous, real-time, high-volume, multiple-source information. On the other hand the rural population of developing countries is under a constant vulnerability before any kind of emergency situations that could have a negative impact in their work activities. Hence it is necessary to develop a system capable of detecting and disseminating automatic alerts in rural zones of developing countries implemented with open source software and low cost hardware.

Considering the previous background, this undergraduate thesis aims at developing a system capable of guaranteeing the early detection and notification of risky and abnormal situations before emergencies via the integration of a complex event processing module inside a converged services platform.

In order to accomplish this objective a characterization of different complex event processing tools and converged services platforms is performed, a set of events patterns which enable the timely detection of risky and abnormal situations is defined, finally an experimental prototype supported by converged environments and CEP which could be leveraged by an early warning system is developed and evaluated.

Some of the obtained the results in the development of this undergraduate thesis are: an integration of the JAIN SLEE specification with the telecommunications services offered by the OpenBTS technology, integration of the JAIN SLEE specification with complex event processing, a implementation of converged services for the deployment of early warnings and a general architecture for the integration of a complex event processing tool and a converged services platform within a conventional telecommunications network.

From the obtained results it can be concluded that the integration of the JAIN SLEE specification, OpenBTS tools and Complex Event Processing is a feasible alternative for the implementation of converged services in rural early warning systems, although the performance of the system can be improved by enhancing the hardware and software tools integrated within the developed platform, since most of the presented inconveniences were caused by the inability to handle large amount of events at the same time, generating the absence of some expected reports during the testing of the experimental prototype.

As a future work it is proposed to realize a performance comparison between our proposal and a similar project implemented with commercial software and hardware tools in order to validate the viability for an actual implementation of a rural early warning system.

Keywords: Converged environments, Complex Event Processing, Early Warning System, OpenBTS, JAIN SLEE Specification, Coffee Rust Detection.

Resumen estructurado

Actualmente los sistemas de alertas tempranas están siendo implementados en entornos convergentes, sin embargo, mejorar la difusión de alertas no es suficiente para este tipo de sistemas, debido a la necesidad de procesar y analizar en tiempo real grandes volúmenes de información proveniente de múltiples fuentes. Por otro lado, la población rural de los países en vía de desarrollo se encuentra bajo la amenaza constante de cualquier situación de emergencia que pueden tener un impacto negativo en sus actividades laborales. Por lo tanto es necesario desarrollar un sistema capaz de detectar y difundir alertas automáticamente en las zonas rurales de los países en vía de desarrollo implementados con software de código abierto y hardware de bajo costo.

Teniendo en cuenta estos antecedentes, este trabajo de grado tiene como objetivo desarrollar un sistema capaz de garantizar la detección temprana y la notificación oportuna de situaciones anormales y de riesgo ante emergencias, a través de la integración de un módulo de procesamiento de eventos complejos dentro de una plataforma de servicios convergentes.

Para lograr este objetivo se lleva a cabo una caracterización de las diferentes herramientas de procesamiento de eventos complejos y plataformas de servicios convergentes, se define un conjunto de patrones de eventos que permiten la detección oportuna de situaciones anormales y de riesgo y finalmente se desarrolla y evalúa un prototipo experimental integrado por entornos convergentes y procesamiento de eventos complejos el cual complementa un sistema de alertas tempranas.

Algunos de los resultados obtenidos en el desarrollo de este trabajo de grado son: la integración de la especificación JAIN SLEE con los servicios de telecomunicaciones que ofrece la tecnología de OpenBTS, la integración de la especificación JAIN SLEE con el procesamiento de eventos complejos, una implementación de servicios convergentes para el despliegue de alertas tempranas y una arquitectura general para la integración de una herramienta de procesamiento de eventos complejos y una plataforma de servicios convergentes dentro de una red de telecomunicaciones convencional.

De los resultados obtenidos se puede concluir que la integración de la especificación JAIN SLEE, OpenBTS y el procesamiento de eventos complejos es una alternativa viable para la implementación de servicios convergentes en los sistemas rurales de alertas tempranas, aunque el rendimiento del sistema puede ser mejorado mediante la actualización de los componentes hardware y software integrados dentro de la plataforma desarrollada, ya que la mayoría de los inconvenientes presentados fueron causados por la incapacidad para manejar una gran cantidad de eventos al mismo tiempo, lo cual provocó la ausencia de algunos informes esperados durante las pruebas del prototipo experimental.

Como trabajo futuro, se propone realizar una comparación de rendimiento entre nuestra propuesta y un proyecto similar implementado con herramientas software y hardware comerciales con el fin de validar la viabilidad de una implementación real de un sistema rural de alerta temprana.

Palabras Clave: Entornos Convergentes, Procesamiento de Eventos Complejos, Sistema de Alertas Tempranas, OpenBTS, Especificación JAIN SLEE, Detección de la Roya en el Café.

Content

Figure list

Table list

Chapter 1

Introduction

1.1. Context

Throughout history, mankind have always been vulnerable before all kinds of risky situations from natural hazards such as earthquakes, avalanches, tornadoes, hurricanes, wildfires to man-made emergencies like chemical and biological agents, hazardous materials, nuclear explosions, among others. With this vulnerability in mind, efforts have been made in the development of tools known as data collection mechanisms. Each of these mechanisms are focused on a specific type of emergency and its primary function is the gathering of context information, i.e. information that enables the prediction of the focused emergency through the analysis of relevant parameters like climatic data (temperature, humidity, wind velocity, etc.). Although some proposals and projects have focused on the automatic detection of emergencies through the analysis of the gathered information, this process has been mainly manual, since a person must look for any possible abnormalities, to subsequently report the situation to the local authorities with any communication mechanism available, from smoke signals to mobile phone calls or even tweets [1].

Considering the vulnerability of mankind, special entities called emergency managers were created, these organizations must be ready to react before any disaster aiming to reduce the negative impact of the aforementioned hazards. However this reactive paradigm is becoming obsolete since the emergency managers only start acting after the occurrence of a disaster, delaying the mitigation of the emergency-produced negative consequences; therefore these entities must embrace new approaches and exploit the advantages offered by modern technologies in order to update the emergency management process and decrease their response time before emergency situations [1].

1.2. Motivation

Currently there is a trend where the policies and efforts implemented by the emergency managers are being focused on decreasing their response time before emergency situations. With this objective in mind, there have been different methods proposed, among which the 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 which provide 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 users in a matter of seconds [1]. Otherwise, between the telecommunications services the mobile services might be the most useful for emergency management, by enabling the possibility to communicate the status of risky situations to a large amount of people within a coverage area [2].

Considering the advantages offered by the telecommunications and Web 2.0 services to the emergency management process, various projects and tools have been developed aiming the early detection of risky situations in different areas. In the health scope the projects have focused on monitoring and detecting irregularities in the patients' health; based on different parameters such as vital signs, these projects can report the patients' current health status to the attending physicians in order to help their decision making and improve their response time. On the other hand, in the environmental scope there have been projects focused on different natural disasters such as avalanches, earthquakes or floods; by monitoring several meteorological parameters relevant to the aimed disaster, these projects can detect atypical situations in order to subsequently notify the threatened population and the emergency managers, decreasing the number of victims and the negative consequences provoked by natural disasters [3].

These examples evidence that early warning systems must react according to the context information in order to successfully detect and notify abnormal situations; this context data can be properly acquired in different manners, but the best way to analyse it is by implementing event-driven software systems, since the data must be monitored as events in order to detect certain patterns which represent the occurrence of an abnormal situation and thereafter perform the required actions [3] [4]. Among the most popular technologies for the development of event-based systems there is the complex event processing (CEP), a technology focused on processing and analysing asynchronous, real-time, high-volume, multiple-source information in order to accurately detect events patterns, allowing the system to react and report any incident as soon as it is detected [5].

1.3. Problem Definition

The early warning systems are currently being implemented inside converged environments due to the aforementioned advantages, however an improved alert dissemination is not enough for this kind of systems due to the necessity of processing and analysing asynchronous, real-time, high-volume, multiple-source information. Therefore an early warning system should implement event-driven software by integrating an event processing methodology, which would improve its data processing and analysing functionalities in order to successfully detect the occurrence of abnormal or risky situations [3] [6].

On the other hand, there is a common problematic in the rural zones of developing countries such as Colombia. This problematic can be defined as a technological deficiency in the telecommunications services like mobile telephony and mobile data, since the farmers and the people that work in the rural zones are outside the coverage area provided by the Telecommunications Service Providers (TSP). This is a common happening in developing countries since the TSP cannot find a profitable outcome by offering its services in low population zones, therefore their services are offered only in urban zones forgetting the countryside. Furthermore the absence of support and investment resources represent the main obstacles for the implementation of any kind of large-scale solution that could resolve the growing issues of the countryside.

The previous scenario illustrates how the rural population of developing countries is under a constant vulnerability before any kind of emergency situations that could have a negative impact in their work activities; for example wildfires, droughts and snowfalls are disasters that directly destroy crops, households or even lives, also the plagues and diseases that affect the crops usually decreasing the production and finally provoking a drop in the population’s quality of life. Hence it is necessary to develop a system capable of detecting and disseminating automatic alerts in rural zones of developing countries implemented with open source software and low cost hardware; such system could be achieved through an integration of CEP and converged environments within an early warning system structure, which leads us to the following research question:

How to guarantee the early detection and notification of abnormal and risky situations through the complex event processing within converged environments?

1.4. Objectives

Considering the previous motivation and the research question, this undergraduate thesis aims at developing a system capable of guaranteeing the early detection and notification of risky and abnormal situations before emergencies via the integration of a complex event processing module inside a converged services platform.

In order to accomplish this objective a characterization of different complex event processing tools and converged services platforms will be performed, a set of events patterns which enable the timely detection of risky and abnormal situations will be defined, finally an experimental prototype supported by converged environments and CEP which could be leveraged by an early warning system will be developed and evaluated.

1.5. Contributions

In the present undergraduate thesis the following contributions can be stated:

Integration of the JAIN SLEE specification with the telecommunications services offered by the OpenBTS technology, providing the possibility of deploying converged services within a conventional telecommunications network.

Integration of the JAIN SLEE specification with complex event processing, providing the converged environments with real-time analysis functionalities, which can be used to keep a context awareness.

Implementation of converged services for the deployment of early warnings through conventional telecommunications and Web 2.0 services.

A general architecture for the integration of a complex event processing tool and a converged services platform within a conventional telecommunications network as part of an early warning system, aiming to improve the emergency management process, illustrated in Figure 1.1 and briefly explained as follows:

> *Converged control module:* This module represents the first part of the Dissemination & Communication element of an early warning system by handling the control and signaling operations needed by the converged services and it is composed by the service container, the management component, the framework component and the resource adapters.

CEP module: This module represents the Monitoring & Warning Service component of an early warning system by handling all the complex event processing operations, i.e. it performs a real time analysis of the incoming information to subsequently generate an automatic warning if an abnormal situation is detected. It is composed by the CEP engine, the event stream adapter, the historical data access and the output adapters.

Telco control module: This module represents the second part of the Dissemination & Communication element of an early warning system by handling the control and signaling operations (communication establishment, switching, ending, etc.) needed to run the telecommunications services, it is composed by service support tools which permit the deployment of one or more traditional Telco services like voice call or SMS.

Business support systems: The support system represents the third part of the Dissemination & Communication element of an early warning system by fulfilling all the features necessary to store the subscribers' information (farmers and local authorities' personal information like phone number and email address) and configuration parameters of the network.

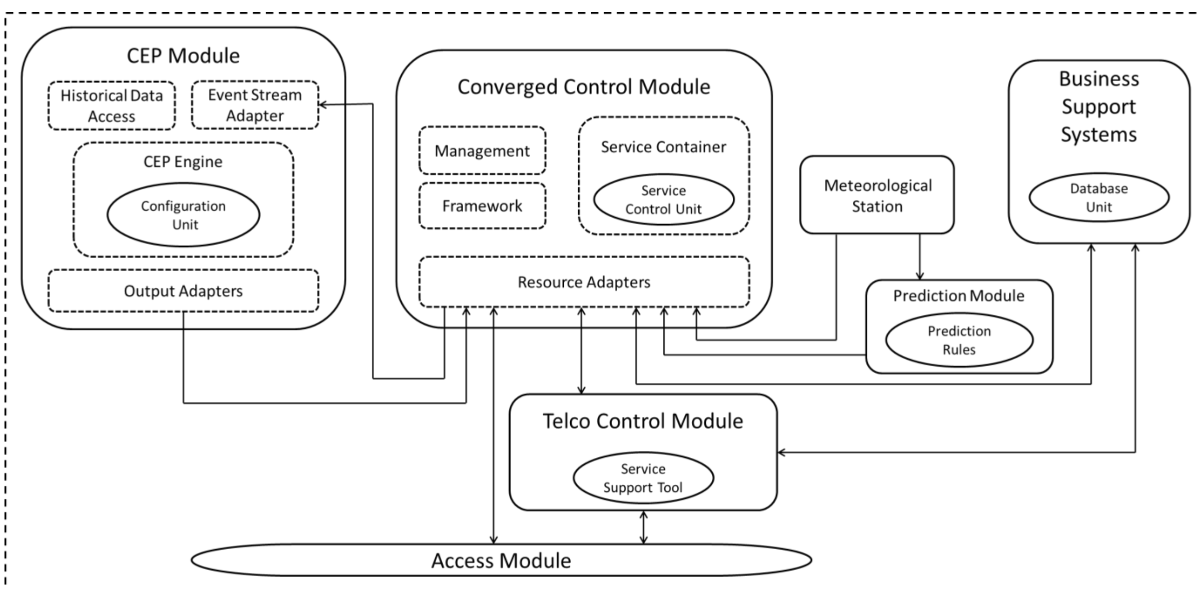

Figure 1.1. Integration architecture.

Access module: This module represents the fourth and last part of the Dissemination & Communication element of an early warning system by referring to all the transport nodes within the integration architecture and acting as an interface which allows accessing the system functionalities.

Meteorological station: The station collects climate information from the sensors network about the four parameters (relative humidity, temperature, wind speed and pluviosity), then every fifteen minutes the collected information is sent via HTTP to the components of the Monitoring and Warning Service of an early warning system (the Prediction Module and the Converged Control Module).

Prediction module: This module is a part of the Monitoring and Warning Service component of an early warning system, it receives the climate information from the meteorological station and analyses it in order to predict the appearance of coffee rust, when this module detects a real threat it sends a report to the Converged Control Module via HTTP.

This architecture will be properly explained in the subsequent sections.

Publishing papers: Within the present undergraduate thesis the following papers were accepted to be published: The first article titled “An Architecture Based On JAIN SLEE and OpenBTS for Rural Early Warning Systems” [7] was presented on the VII Congreso Iberoamericano de Telemática [8] conducted in Popayán, Colombia from 10 to 12 June of 2015, this paper describes the implementation and evaluation of the integration between the JAIN SLEE specification and OpenBTS tools. The second paper titled “Improving Rural Early Warning Systems Through The Integration of OpenBTS and JAIN SLEE” [9] was presented to the *Revista Ingenierias Universidad de Medellín* (ISSN 1692-3324) and is available in the number 29 of the journal [10].

1.6. Content

The structure of the present undergraduate thesis will be described as follows:

Chapter 2: This chapter presents the state of the art and related works of early warning systems involved with converged environments and complex event processing, also the characterization and selection process of the different tools needed in order to accomplish the stated objectives is described.

Chapter 3: This chapter presents a detailed description of the application scenario for this undergraduate thesis, considering a specific case study focused on the coffee rust early detection and warning. Besides, the definition of the different event patterns implemented within the experimental prototype is exposed. Finally the integration architecture of CEP with converged environments is presented.

Chapter 4: This chapter presents a detailed description of the integration between OpenBTS within converged environments aimed at rural zones of developing countries. Furthermore the service catalog implemented within the platform is explained and finally an application example of the system is illustrated.

Chapter 5: This chapter presents the experimental prototype of the platform implemented within this undergraduate thesis, starting with the deployment model of the experimental prototype and its components; in the second place the experimental prototype is tested in three different ways, presenting the tests' objectives, methodology and results.

Chapter 6: This chapter presents the conclusions obtained from the development of this undergraduate thesis along with some possible future works.

Chapter 2

State of the art

This chapter introduces a description of the most relevant concepts and technologies within this undergraduate thesis, the explained concepts are: converged environments, the JAIN SLEE specification, OpenBTS, complex event processing and early warning system; subsequently presenting a set of related works about the deployment of warnings in converged environments, dissemination of converged services through OpenBTS and complex event processing in converged environments; and finally exposing the characterization and selection of the converged environments platforms and complex event processing tools, following a set of specific criteria.

2.1. General context

Before describing the solution proposed by this undergraduate thesis it is necessary to define and comprehend some of the most important concepts, components and technologies surrounding the JAIN SLEE specification, OpenBTS, CEP and Early Warning systems, which will be briefly described as follows.

2.1.1. Converged environments

The Web 2.0 services and telecommunications services can be very useful for the early warning systems, the first ones by sharing enormous amounts of multimedia information like videos and pictures to hundreds of thousands users in a matter of seconds [11], the second ones by enabling the possibility to communicate the status of risky situations to a large amount of people within a coverage area [3].

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 the concepts, technologies and services of the Web 2.0 with the 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 complexity.

There are a few technologies which enable the development of converged services, where the JAIN SLEE specification stands out as one of the most robust and complete possibilities.

JAIN SLEE Specification

The JAIN SLEE specification is composed by two concepts which might be analysed independently. Firstly is the SLEE (Service Logic Execution Environment), this environment 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, also these capabilities 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 to be analysed is the JAIN (Java API for Integrated Networks) standard. This standard defines a set of Java API aimed at developing telecommunications 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 [12] [13].

Considering the analysis of these concepts, it is not difficult to understand how the JAIN SLEE [14] 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 [12].

2.1.2. Open Base Transceiver Station

OpenBTS (Open Base Transceiver Station) 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 (Global System for Mobile Communications) 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 [15], 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 [16].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 [17].

2.1.3. Complex Event Processing

Considering that the events represent an important source of information about the context of a system, there have been different event processing methodologies developed that have allowed building event-driven software systems. These systems are composed by three main components: a monitoring component responsible for the observation of events and its representation; a transmission mechanism that performs event notification by different means; and a reactive component, i.e. which actions the system will take once the event or pattern of events is detected [3].

Among the most popular technologies for the development of event-based systems there is the complex event processing (CEP). This technology is focused on processing information from multiple sources so it is possible to detect patterns of events accurately, allowing the system to react and report the incident as soon as it is detected [5].

2.1.4. Early Warning System

In the 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 [18].

A complete and effective early warning system comprises four key elements: Risk Knowledge, a component which allows the system to assess the actual risk of a situation due to its emergency related collected information; 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; Dissemination and 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; Response capability, this component represents the knowledge and response mechanisms the population has, i.e. how the people must react before a dangerous situation and which tools they have to face the phenomenon (escape routes, first aid kits, etc.) [18].

2.2. Related works

This section presents an analysis of the most important related works within the framework of this proposal, highlighting that until now there have not been projects that consider a similar or identical approach to the one presented in this undergraduate thesis. Therefore, the current state of knowledge will be presented focusing on the deployment of warnings in converged environments, the dissemination of converged services through OpenBTS and complex event processing in converged environments as the most relevant areas to the domain this proposal.

2.2.1. Deployment of warnings in converged environments

In the health scope a right decision-making is vital for patients' care, especially in the intensive care units. However, keeping a proper control on every patient of a large hospital is a rather complex task due to the large number of people who is under care. In order to maintain a constant monitoring on patients and improve the decision-making, in 2008 Van Den Bossche and his team developed a platform supported by JAIN SLEE which enables the reception of large amounts of information about the current status of patients, analyse it according to predetermined levels and establish a communication with the physicians in case of any abnormality; the communication is established through different web and telecommunications services as text messages or e-mail [19].

Taking into account the new paradigm of emergency management that is being implemented by some emergency managers, where civilians change their typical role of victims to a more active one as informants, in 2011 Ullah Khan and his team developed an infrastructure for emergency reports based on a repository of information that is divided into two large scale databases. The first one is a database based on the information obtained from the emergency managers and public institutions such as hospitals and universities which report about any event through traditional mechanisms. The second database is focused on obtaining multimedia information from the most popular social networks, i.e. information obtained from the users who report risky situations through their Facebook or Twitter accounts, leveraging the social networks capacities on information distribution. The combination of these databases enables a more active interaction between civilians and emergency managers, generating a constant monitoring by the community and decreasing the response time before emergencies; since once a risky situation is identified the system proceeds to notify the emergency managers through converged services, so they can proceed with the necessary procedures [20].

The development of a platform for the automated composition of converged services aimed at environmental early warnings constitutes a current paradigm (2014). From this approach emerged the system known as AUTO [21], which has been designed aiming rural areas of developing countries where the population has no training in handling highly complex systems and the technology is generally limited and outdated. AUTO enables the users to perform a service composition from natural language through a module capable of recognizing patterns and keywords in the requests, it also implements various sensor networks in order to maintain a constant environmental monitoring and establish a context awareness concerning the users' surroundings; this context awareness enables the system to automatically generate the most suitable converged service based on the user needs, considering their requests and context [21].

2.2.2. Dissemination of converged services through OpenBTS

The mobile services have showed a vertiginous advance since its first commercial deployments, firstly providing basic connectivity for voice communication services, and currently becoming a convergence of affordable services from smart terminals, enabling the access to multimedia content through a high quality Internet connectivity. However, although the supply of this type of mobile services is a reality in many developed countries, the infrastructure that must be implemented to support them demands a large investment from the operators, which leads them to question themselves whether such investment would be profitable in a particular area, therefore the TSP generally decide to offer their services only in urban areas with a large population, depriving small urban communities and rural zones of their services.

Since the absence of communication networks is a clear sign of underdevelopment, in recent years several teams have designed systems that enable the deployment of low-cost base stations for rural zones. Among these researches stands out Hajar's project in Lebanese Republic, focused on reducing the installation costs of the Base Transceiver Stations (BTS) to a minimum by implementing low cost technologies like OpenBTS and Raspberry Pi [22]. Another development based on reducing the BTS operating costs is Heimerl's project in the United States, Heimerl and his team designed a more expensive system infrastructure that works with its own solar power plant, which has low operating costs thanks to a mechanism that turns off the BTS when it has not been used by any user for a certain amount of time and turns it on again when a user attempts to access its services [23]. Besides, there have been experiments in rural areas, like the one performed by Van Stam and Mpala in Zambia, they implemented a low-cost mobile network using OpenBTS, enabling internet access, in a rural zone of Zambia. Through this implementation they were able to establish voice calls to and from mobile phones in Zambia maintaining low installation and maintenance costs [24].

There are several works that have focused on providing GSM coverage in rural zones, however, through a 2.5G network it is difficult to provide advanced telecommunications services; given this situation, in 2013 a team led by Raja Ali Anwaar developed a project aiming to provide a low cost 2.5G network with some capabilities of a 3G / 3.5G network by leveraging the OpenIMSCore and OpenBTS technologies. In a more precise way, they implement a 2.5G network through the OpenBTS software and hardware components, where the Universal Software Radio Peripheral (USRP) provides network access and the Asterisk PBX software performs the switching operations. Once the GSM network was established the Asterisk functionalities were replaced by OpenIMSCore, equipping the network with an IP Multimedia Subsystem (IMS); this equipment enabled a better quality in the Internet services provision and a more advanced multimedia content without changing the way of accessing the network and maintaining low costs in infrastructure investments [25].

2.2.3. Complex event processing in converged environments

Complex event processing is a tool that enables the analysis of different aspects of society such as health, emergency response, safety and entertainment, among others; by designing new monitoring systems and improving the existing ones. In the health scope the main objective is the welfare of patients, so it is vitally important to perform an early identification of symptoms that may represent a health risk. To achieve this goal, in 2009 Foley and Churcher integrated complex event processing on an existing platform developed by a group of British universities called SAPHE (Smart and Aware Pervasive Healthcare Environment), this platform performs a constant monitoring of patients while they stay in their homes. Originally SAPHE consisted of a set of sensors deployed in both the patients' bodies and their homes, communicated through wireless technologies such as Bluetooth and ZigBee with an information receptor that stored data related with eating habits, blood pressure, sleep disturbances, heart rate, among others, such information was subsequently sent to the physicians who maintained a surveillance in the patient's status. When complex event processing was integrated within the platform, the collected information stopped being studied by health professionals to become internally analysed by the system. This change resulted into an improved detection of situations that could affect the patients' welfare, besides a better efficiency while processing large volumes of information [6].

That same year, Hinze's team in Germany, noting the growing trend in the use of event processing for the developing of monitoring applications, conducted a project with four specific objectives: define the basics of event processing and create a common understanding throughout various application examples; identify technologies related to event processing; describe a wide range of applications and their main functions; finally identify problems and research topics that are common in different event processing domains [3]. Through this research various projects supporting the fact that the event processing can be applied in different scenarios were identified, among which stand: a tourism system in charge of identifying the user's position through its mobile phone and then displaying the location of nearby tourist attractions on Google Maps; an avalanche prevention system developed in Switzerland which by combining information gathered by a sensor network and weather forecasts, notifies the population about the possible risk of a flood through telecommunications services; a system to prevent fraud on credit cards and mobile devices, which by identifying usage patterns of the original holders can detect suspicious activities, e.g. when the account holder is not the only one who is currently using the credit card the system could determine that the management of such account is abnormal, proceeding to report a possible impersonation fraud through various mechanisms to both the original account holder and the financial service provider [3].

Later, in 2011, the Fujitsu researchers Kobayashi and Fujita proposed the idea of a “Human-Centric Intelligent Society” by leveraging the usefulness of ICT in daily life [4]. The objective of this proposal is to monitor and improve the quality of life of the people inside a community through the collection and analysis of large amounts of information about the community’s context. An application example proposed by the researchers is the improvement of the Japanese taxi service through the analysis of vehicles’ location, time and routes, predicting the formation of traffic jams and recommending better routes to the taxi drivers so they can offer a better quality of service. This type of converged services could be acquired from the Convergence Service Platform, an ICT platform composed of twelve subsystems, highlighting the module for processing complex events which allows to analyse the collected information in order to subsequently predict and prevent different situations depending on the community’s context [4].

2.2.4. Gaps

The table 2.1 summarizes the gaps found in the related works. From the projects related with the deployment of warnings in converged environments Ordonez work [21] stands out, which is focused on the composition of converged services specially designed for the deployment of environmental alerts in rural areas; however, despite considering environmental monitoring, it does not use complex event processing to analyse such information neither takes advantage of OpenBTS to provide mobile coverage in areas that do not have mobile services. On the other hand, regarding the projects related with the dissemination of converged services through OpenBTS the work of Anwaar Ali [25] highlights, this project succeeds in making an implementation of OpenIMSCore within an established OpenBTS network, which equips the network with an IP multimedia subsystem that allows it to provide services comparable to those of a 3G / 3.5G network, with the advantage that the necessary investments for infrastructure and maintenance are considerably lower than the investments needed for traditional networks; although this is a step in the dissemination of converged services via OpenBTS, this network only offers traditional telecommunications services without a real integration of converged services and complex event processing, thus integrating OpenIMSCore with OpenBTS is not considered as a primary objective within this project's framework.

Deployment of warnings in converged environments		
Related work	**Contributions**	**Gaps**
Van Den Bossche et al. [19]	- Integration of a monitoring system within a converged service platform. - Performs an effective deployment of warnings in case of emergency.	- Does not implement complex event processing for information analysis. - Does not consider the implementation of a low-cost telco network. - Does not consider the deployment of warnings in rural areas.
Li et al. [20]	- Integration of a monitoring system within a converged service platform. - Performs an effective deployment of warnings in case of emergency. - Leverages the social networks as an information source.	- Does not implement complex event processing for information analysis. - Does not consider the implementation of a low-cost telco network. - Does not consider the deployment of warnings in rural areas.
Ordoñez et al. [21]	- Integration of a monitoring system within a converged service platform. - Performs an effective deployment of warnings in case of emergency. - Automated service composition through pattern recognition in common language.	- Does not implement complex event processing for information analysis. - Does not consider the implementation of a low-cost telco network.

Dissemination of converged services through OpenBTS		
Related work	**Contributions**	**Gaps**
Hajar et al. [22]	- Implementation of the OpenBTS technology to offer GSM coverage in rural areas.	- Does not implement complex event processing for information analysis. - Does not consider the deployment of warnings in rural areas. - Does not consider the implementation of converged services
Heimerl et al. [23]	- Implementation of the OpenBTS technology to offer GSM coverage in rural areas.	- Does not implement complex event processing for information analysis. - Does not consider the deployment of warnings in rural areas. - Does not consider the implementation of converged services.
Mpala and Van Stam [24]	- Implementation of the OpenBTS technology to offer GSM coverage in rural areas.	- Does not implement complex event processing for information analysis. - Does not consider the deployment of warnings in rural areas. - Does not consider the implementation of converged services.
Ali et al. [25]	- Implementation of the OpenBTS technology to offer GSM coverage in rural areas. - Enabling the offer of internet services similar to 3G/3.5G in a GSM network.	- Does not implement complex event processing for information analysis. - Does not consider the deployment of warnings in rural areas. - Does not consider the implementation of converged services.
Complex event processing in converged environments		
Related work	**Contributions**	**Gaps**
Foley and Churcher [6]	- Integration of CEP functionalities within an existing monitoring system. - Implementation of converged services for information distribution.	- Does not consider the implementation of a low-cost telco network. - Does not consider the deployment of warnings in rural areas.
Hinze et al. [3]	- Offers a general vision of the complex event processing concept and its different application areas. - References different projects that consider the analysis of various emergencies. - Highlights the implementation of converged services in different projects.	- Does not consider the implementation of a low-cost telco network.
Kobayashi and Fujita [4]	- Presents the concept of a "Human-Centric Intelligent Society". - Proposes an architecture for a converged service platform. - Implements complex event processing for information analysis. - Enables trends identification based on the information analysis.	- Does not consider the implementation of a low-cost telco network. - Does not consider the deployment of warnings in rural areas.

Table 2.1. Identified gaps.

Finally, regarding the projects related with complex event processing in converged environments the work of Kobayashi and Fujita [4] stands out, through the concept of an "Human-Centric Intelligent Society", the researchers propose an architecture for a converged service platform having two main objectives: Firstly the collection and analysis of user information through sensor networks and mobile devices in order to identify trends which allow predicting the users' behavior; Secondly the recommendation of appropriate services to the users, based on the analysed context information. These objectives are achieved through complex event processing and converged services, however, in their proposal they do not include the monitoring of emergency situations neither the integration of OpenBTS based networks.

2.3. Characterization and selection

This section presents the characterization and selection process of the different technologies implemented within this undergraduate thesis. The selection process of the telecommunications service dissemination tool is presented in the first place, following with the selection process for the converged service development and deployment platform and concluding with the selection process for the complex event processing tool.

2.3.1. Selection process for the telecommunications service dissemination tool

A brief characterization of the considered tools for the dissemination of telecommunications services is presented in this subsection, followed by the description of the evaluation criteria and the obtained results from the performed evaluation.

Tools characterization

The different tools for the telecommunications service dissemination are presented as follows:

Open Base Transceiver Station

OpenBTS is an open source project whose main objective is to generate an infrastructure capable of establishing a GSM mobile connection without the need of a traditional TSP, saving installation costs and offering a completely customizable mobile station. This application is composed by a GSM access point based on software that allows its SIP PBX to detect any GSM mobile phone as SIP terminals, as if they were in a VoIP network, enabling the replacement of the traditional networks for low cost and easy installation stations, where every user of this new network can connect with any other user on the same network or with other networks' users through Internet. It should be noted that mobile phones continue operating under the GSM technology, through an OpenBTS system, not a traditional TSP [16].

Among the resources required for the installation of OpenBTS stations there are software and hardware tools such as: the Universal Software Radio Peripheral (USRP), used to present a GSM interface using the GNU Radio framework; the open source PBX Asterisk, used for the switching and control of voice calls; the OpenBTS application; and a computer with Linux operating system [17]. Although there is a considerable amount of information about the usage of OpenBTS, such information is scattered and disorganized, hence the amount of time required to install the basic functionalities of OpenBTS increases. Therefore the Appendix A offers a step by step installation guide which gathers information from multiple sources.

Femtocells

A femtocell is a small base transceiver station with limited coverage and power, designed to be implemented in homes or small businesses. The femtocells are usually connected to the service provider's network through a wired broadband like ADSL (Asymmetric Digital Subscriber Line). A femtocell allows a TSP to improve its indoor coverage, and also enables near base transceiver stations to release communication channels. The femtocell incorporates the functionalities of a typical base station but extends it to enable a simpler and autonomous deployment [26].

Base transceiver stations

A base transceiver station (BTS) has transmission/reception radio equipment under the 850/900/1800/1900 MHz frequency bands for GSM and 1900/2100 MHz for UMTS. These stations are responsible for establishing the communication links when a user makes or receives a call or a message with a mobile phone. The implemented antennas are usually in the top of towers, buildings or hills to provide better coverage. Besides, the base transceiver station generally has a radio or wired transmission mechanism to maintain a communication with the PBX, which performs the routing of the calls to the destination phone [27].

Evaluation criteria

The evaluation criteria implemented for the selection process of the telecommunications service dissemination tool are explained subsequently:

Tool Availability: This criterion refers to the monetary investment required to acquire the tools needed for the implementation of a network capable of performing the dissemination of telecommunications services. This criterion is relevant to the tool selection because this module could be the most expensive one among all the project components.

Open Source: This criterion represents the freely distributed and developed software, i.e. software that can be used for any purpose, it can be studied, modified, improved and distributed freely. This feature is important for the development of this undergraduate thesis since it reduces costs and simplifies its time to market, besides enables a detailed understanding of the tools operation in order to guarantee interoperability [28].

Developers community: This criterion refers to the group of developers who use a specific software or hardware tool. A developers community is qualified according to the amount of information of experiences and results obtained in developed projects shared by all developers. This feature is important for this undergraduate thesis since the feedback obtained from the experiences of other developers enables an enhanced tool employment.

Documentation: The documentation is the official available information concerning the implementation of a tool, i.e. the information offered by the tool provider or any qualified entity. This feature is important for this undergraduate thesis since the efforts needed for the implementation of any tool are simplified by having a good documentation.

Evaluation and analysis

The table 2.2 illustrates the obtained results in the performed evaluation for the selection of the telecommunications service dissemination tool, the implemented scale is from 0 to 5, with 0 being the worst score and 5 the best score, giving each criterion a percentage according to its importance.

Tools	Availability (30%)	Open Source (30%)	Community (20%)	Documentation (20%)	Results
OpenBTS	5,00	4,00	4,50	4,00	4,40
Femtocells	4,00	2,00	3,00	4,50	3,30
Base Transceiver Stations	0,00	0,00	1,00	5,00	1,20

Table 2.2. Selection of the telecommunications service dissemination tool.

As observed from the table 2.2, according to the obtained results from the performed evaluation, the tool that properly adjusts to the conditions of this undergraduate thesis is OpenBTS. It should be noted that while this tool is more expensive on the market than a femtocell, the University of Cauca provided the necessary hardware for the project.

2.3.2. Selection process for the converged service development and deployment platform

This subsection, based on [12], presents a brief characterization of the considered platforms for the development and deployment of converged services, followed by the description of the evaluation criteria and the obtained results from the performed evaluation.

Platforms characterization

The different platforms for the converged service development and deployment are presented as follows:

Oracle

The rapid deployment of telecommunications services using open standards is the concept that has adopted Oracle for their products aimed at the telecommunications sector [29]. The Oracle Communications Server is composed by a set of products that enable the creation, execution, presentation and management of telecommunications services. The converged communications application server and the converged communications GateKeeper are the two main products of this company for the development of converged services in Next Generation Networks (NGN) [30]. Both products are described as follows:

Converged communications application server: is based on Java Enterprise Edition (JavaEE) and supports the SIP protocol through SIP Servlets. Furthermore it is built under the Service Oriented Architecture (SOA) paradigm and is a powerful tool with high levels of availability, reliability and performance, which allows it to provide a variety of converged services in NGN environments that have an IMS-based control layer.

Converged communications GateKeeper: is a converged service exhibition platform based on SOA, which allows the network operators to potentiate their services by linking new Application Service Providers (ASP) that enhance the contents provided to the users. It includes all the security mechanisms required by an operator to expose its network capabilities and performs processes related with the quality of service (QoS) needed to meet the high requirements needed to provide converged services.

IBM

IBM developed an environment for the TSP capable of offering a variety of converged services known as SPDE (Service Provider Delivery Environment), which provides mechanisms for service creation, security, management, maintenance and deployment. Among the most important features of SPDE are [31]: horizontal integration of business models, based on IT and telecommunications standards (SOA, Web 2.0, eTOM, NGOSS, IMS , SIP, etc.), service platform independent from the TSP network, support to multiple service creation environments, network abstraction and exposure to third parties through API and Web services, loose coupling between software components and adaptability.

Ericsson

Within its three working areas (networks, services and multimedia), Ericsson defined a multimedia platform for the deployment of advanced services in mobile telecommunications networks. Among the most important features of this platform are: service delivery support, orchestration and business process, support for common functions and service execution [32].

Red Hat

Years ago the company completely acquired Mobicents, an open source tool for the implementation of the JAIN SLEE specification. With the combination of JBoss and Mobicents server, Red Hat has created an integrated platform for the creation, implementation and deployment of converged services in the telecommunications industry, supported on IT technologies like SOA [33].

Open Cloud

RHINO, the commercial distribution of this company's JAIN SLEE platform, defines the most important components of a service deployment platform, its complete structure is formed by the service creation, execution, integration and abstraction level layers [34].

RHINO JSLEE: The main component of the RHINO platform is composed by the application server, designed and optimized for event-driven communication applications, supporting the requirements of a TSP.

RHINO SIS (RHINO Service Interaction Server): A powerful, flexible and extensible integration service platform handled by scripts and user interfaces. This component allows the developer to compose and manage new network services using the Signaling System No. 7 (SS7) and IMS.

RHINO RA: Provides a set of technologies that are introduced into the JAIN SLEE platform to interoperate different systems, including resource adapters (RA) for SIP, ISC (3GPP / Extension Support), Diameter (IMS Interface), MM7 (Multimedia Messaging), SS7, Simple Object Access Protocol (SOAP), JavaEE, Lightweight Directory Access Protocol (LDAP), Java Data Base Connectivity (JDBC), among others.

Evaluation criteria

The evaluation criteria implemented for the selection process of the converged service development and deployment platform are explained subsequently:

OSA/Parlay support: the purpose of the Parlay group from the Open Services Access (OSA) initiative is to offer a set of API to create new services regarding the underlying network and programming technology, hence the OSA/Parlay API are specified in Unified Modeling Language (UML) establishing a set of specifications for programming environments like Common Object Request Broker Architecture / Interface Definition Language (CORBA/IDL), Java and Web Services Description Language (WSDL). Among the most important Parlay API are: call control, conferences, text and audio messages, and billing; these API are designed to facilitate the creation of services by organizations both within and outside the TSP environment [35].

The OSA/Parlay support is important for this undergraduate thesis since it is necessary for the converged services platform to support API that enables calls control and text messages.

IMS compatibility: the IP Multimedia Subsystem is a set of specifications that describe how the NGN architecture supports telephony and multimedia services through IP. More precisely, IMS defines a framework and architecture for voice, data, video, images and services through an infrastructure based on packet routing through IP addresses. IMS compatibility is important for this undergraduate thesis since it allows to integrate voice, multimedia and data services in any network platform accessible through an internet connection [36].

Fixed and mobile networks support: The capacity to provide fixed and mobile networks support is an essential feature in a telecommunications server, besides considering this undergraduate thesis framework, it is important for the integration of the deployed converged services with the mobile network supported by the OpenBTS tool.

OSS/BSS support: the Operation Support Systems (OSS) are computer systems implemented by TSP to manage their networks. OSS usually describes the network systems that are directly linked to the telecommunications network itself, for example, support processes for maintaining network inventory, provisioning services, network elements configuration and software fault management. On the other hand the Business Support Systems (BSS) are the components that a TSP uses to run its business operations, i.e. customer care, support processes and billing [37]. The OSS/BSS support is important for this undergraduate thesis since these systems are key elements of any telecommunications server.

Supported protocols: considering this undergraduate thesis framework, the protocols that the telecommunications server must support to fulfill the objectives are:

> *SIP:* the Session Initiation Protocol is an application-layer control protocol capable of establishing, modifying, and terminating multimedia sessions such as IP Telephony calls, i.e. SIP can establish a session between two or more users, invite new participants into this session and add or remove multimedia information from it. SIP transparently supports name mapping and redirection services in order to support personal mobility, this way users can maintain a single externally visible identifier regardless of their network location [38].
>
> *HTTP:* the Hypertext Transfer Protocol is a transaction-oriented protocol and follows the request-response scheme between a client and a server. The customer who makes a request (usually from a web browser) is known as user agent, the transmitted information is called resource and it is identified by a Uniform Resource Locator (URL). The resources can be files, the result of an executing a program, a query to a database, the automatic translation of a document, etc. Supporting the HTTP protocol is important for this undergraduate thesis since it defines the syntax and semantics needed to communicate implemented by the software components of the web architecture (clients, servers, proxies) [39].
>
> *DIAMETER:* is a network protocol for the authentication of users that remotely connect to the Internet through the PSTN (Public Switched Telephone Network). It provides authorization and auditing services for applications such as network access or IP mobility. The basic concept of DIAMETER is to provide a base protocol that can be extended to provide authentication,

authorization and auditing to new access technologies. Supporting this protocol is important for this undergraduate thesis since it allows to offer highly mobile, dynamic, flexible and versatile services [40].

Evaluation and analysis

The table 2.3 illustrates the obtained results from the performed evaluation for the selection of the converged service development and deployment platform.

As shown in the table 2.3 the telecommunications servers that meet all the requirements outlined in the present subsection are Open Cloud's RHINO and Red Hat's Mobicents. In order to choose the telecommunications server that fulfills the requirements of the project it is necessary to consider a different criterion from those defined in the evaluation: having an open source license.

Criterion	Oracle	Ericsson	IBM	Open Cloud	Red Hat
Supported languages	Java	Java	Java	Java	Java
Execution environment	SIP Servlets	SIP Servlets	SIP Servlets	JSLEE (RHINO)	JSLEE (Mobicents)
OSA/Parlay support	NO	NO	NO	YES	YES
IMS compatibility	YES	YES	YES	YES	YES
Fixed and mobile networks support	YES	YES	YES	YES	YES
OSS/BSS support	YES	YES	YES	YES	YES
Supported protocols	SIP HTTP	SIP HTTP	SIP HTTP	SIP SS7 INAP CAP MAP Diameter Others	SIP SS7 INAP CAP MAP Diameter Others

Table 2.3. Selection of the converged service development and deployment platform.

An open source license assures that the software can be studied, modified, improved, used for any purpose and distributed freely. This feature is important for the development of this undergraduate thesis since it reduces costs and simplifies the time to market, besides it enables a detailed understanding of the functioning of the tools facilitating the interoperability [28]. Therefore the server that is best suited to the project's requirements is Red Hat's Mobicents.

2.3.3. Selection process for the complex event processing tool

This subsection presents a brief characterization of some complex event processing tools, followed by the description of the evaluation criteria and finally the obtained results from the performed evaluation.

Tools characterization

The different tools for the complex event processing are presented as follows:

Esper

Esper is an open source component designed for complex event processing and event series analysis, it is compatible with different programming languages through its two distributions: Esper for Java platforms and Nesper for .Net platforms [41]. Esper and Nesper enable an agile and efficient development of applications capable of processing large volumes of events, regardless of whether those events have already happened or are being analysed in real time, besides it enables the filtering and analysis of events in multiple ways in order to successfully meet the requirements set by the developer [41].

Among the most important features of this component are: the event processing language (EPL) which allows to define the conditions to perform an event pattern detection based on Structured Query Language (SQL) statements; a high scalability; compatibility with a variety of medium or high speed data; a Big Data engine with real-time processing and streaming capabilities; and compatibility with development environments such as Eclipse, Netbeans and web servers [41].

One of the main advantages of Esper is its large developers community, enabling the generation of a complete and understandable documentation, so the developers without experience in the field of complex event processing can create complete applications using their guides and manuals. Esper community is open, which also allows it to provide good support to developers who issues when implementing the tool [41] [42].

Microsoft StreamInsight

Microsoft StreamInsight is a platform implemented to develop and deploy applications for complex event processing and it is oriented for Windows only. Its high performance stream processing architecture and the development platform based on Microsoft .NET framework enables the implementation of effective event processing applications. By using StreamInsight, the developers can create CEP based applications that have the ability to gather, analyse and correlate the information, besides the developers are given the capacity to monitor and manage the information observing the conditions, opportunities and faults in an almost instantaneous way [43].

This tool has a complete support and documentation provided by Microsoft, since to implement StreamInsight the developer must have a SQL Server license, however, since it is a proprietary license, the developer community is small and generally closed, resulting in the need to acquire the support offered by Microsoft to its customers [43] [44] [45].

APAMA

The Apama platform, developed by the german software company AG enables organizations to analyse and take decisions in real time based on information related to business operations and interactions with the customers. Apama is only compatible with its own integrated development environment called Apama Studio where applications can be implemented with the capacity to correlate, aggregate and detect patterns through large volumes of data coming from multiple sources, so that decisions can be made at the right time, besides it enables organizations to develop and deploy scalable solutions related to real time intensive analysis and telecommunications [46].

Despite being recognized in 2014 as one of the best tools for complex event processing, Apama has issues regarding the documentation about the tool’s functionality as it is difficult to understand, has no application examples and does not specify any compatibility with servers or programming languages, hence it is very difficult to implement the tool in the development of any separate project [46] [47].

Altibase

Altibase system is a relational database developed by Altibase Corporation that was specially designed to be used by large companies. This tool specializes in "Computer memory" technologies, focusing on high intensity transactional processing through the Random Access Memory (RAM) [48].

Altibase has three main services: Altibase XDB is a database engine in memory, i.e. a database system that stores and processes information in the RAM, speeding up the transactional and analytical processing, reaching a speed of a million transactions per second (TPS); Altibase HDB is a hybrid database engine, i.e. a system that combines the processing speed in memory with the storage capacity of a traditional database; Altibase CEP is its module for complex event processing, also known as DMS Altibase, this tool allows the developers to filter and analyse information from multiple sources in real time while leveraging the advantages of XDB and HDB for a better processing [49] [50].

The support provided by Altibase Corporation to its customers is very complete, hence their services are largely implemented by TSP, specially for the billing purposes, however for this same reason they implement a proprietary license directed to a specific market, therefore its developers community is small and closed. Altibase has a very comprehensive database documentation, however lacks of documentation about their complex event processing service Altibase CEP [50] [51].

Oracle Event Processing

Oracle Event Processing (OEP) is a platform for the creation of event-driven applications, capable of filtering, correlating and processing events in real time, therefore other kinds of applications such as service-oriented and event-driven applications can be monitored by a tool that actually works in real time.

Oracle Event Processing provides organizations with a "top-down" solution aimed at the design, definition, development and implementation of event-driven applications that not only fulfill the previously defined requirements but exceed the expectations. Developed on the latest standards including ANSI SQL (American National Standards Institute - SQL), Java, Open Services Gateway Initiative (OSGi) and Spring DM, OEP provides an open architecture that enables the gathering, processing and publication of complex events across an enterprise. Furthermore Oracle provides a complete and detailed documentation about the tool’s performance and currently there is a large developers community that offers support for the most common difficulties presented while employing the tool [52] [53].

Drools

Drools is a Business Rules Management System solution; it provides a core Business Rules Engine (Drools Expert), a web authoring and rules management application (Drools Workbench), and Complex Event Processing features (Drools Fusion), available via an Eclipse IDE plugin for core development [54]. Drools was also born as a rules engine (JBoss Rules) several years ago, but following the vision of becoming a single platform for behavioral modelling, it soon realized that it could only achieve this goal by crediting the same importance to the three complementary business modelling techniques: Business Rules Management, Business Processes Management, and Complex Event Processing [54].

In this context, *Drools Fusion* is the module responsible for adding event processing capabilities into the platform. Supporting Complex Event Processing, though, is much more than simply understanding what an event is. CEP scenarios share several common and distinguishing characteristics: Usually required to process huge volumes of events, but only a small percentage of the events are of real interest. Events are usually immutable, since they are a record of state change. Usually the rules and queries on events must run in reactive modes, i.e., react to the detection of event patterns. Usually there are strong temporal relationships between related events.
Individual events are usually not important. The system is concerned about patterns of related events and their relationships. Usually, the system is required to perform composition and aggregation of events [55].

Based on this general common characteristics, *Drools Fusion* defined a set of goals to be achieved in order to support Complex Event Processing appropriately: Support Events, with their proper semantics, as first class citizens. Allow detection, correlation, aggregation and composition of events. Support processing of Streams of events. Support temporal constraints in order to model the temporal relationships between events. Support sliding windows of interesting events. Support a session scoped unified clock. Support the required volumes of events for CEP use cases. Support to (re)active rules. Support adapters for event input into the engine (pipeline) [55].

Evaluation criteria

The evaluation criteria implemented for the selection process of complex event processing tool are explained subsequently:

Java support: Considering the converged service development and deployment platform selected in the subsection 2.3.2 (Mobicents JAIN SLEE) implements the Java language, a Java language support is important within this undergraduate thesis since it ensures better interoperability between components.

Open Source: This criterion represents the freely distributed and developed software, i.e. software that can be used for any purpose, it can be studied, modified, improved and distributed freely. This feature is important for the development of this undergraduate thesis since it reduces costs and simplifies its time to market, besides enables a detailed understanding of the tools operation in order to guarantee interoperability [28].

Developers community: This criterion refers to the group of developers who use a specific software or hardware tool. A developers community is qualified according to the amount of information of experiences and results obtained in developed projects shared by all developers. This feature is important for this undergraduate thesis since the feedback obtained from the experiences of other developers enables an enhanced tool employment.

Documentation: The documentation is the official available information concerning the implementation of a tool, i.e. the information offered by the tool provider or any qualified entity. This feature is important for this undergraduate thesis since the efforts needed for the implementation of any tool are simplified by having a good documentation.

Compatibility with Linux OS: Considering the selected tool for the telecommunications service dissemination (OpenBTS) only works on Linux, the compatibility with this operating system is important to ensure a good performance of the system.

Evaluation and analysis

The table 2.4 illustrates the obtained results from the performed evaluation for the selection of the complex event processing tool; the implemented scale is from 0 to 5, with 0 being the worst score and 5 the best score, giving each criterion a percentage according to its importance.

Tools	Java Support (25%)	Open Source (30%)	Community (15%)	Documentation (10%)	Linux Compatibility (20%)	Results
Esper	5,00	4,50	5,00	5,00	5,00	4,85
Microsoft StreamInsight	0,00	2,00	3,00	5,00	0,00	1,55
APAMA	2,00	0,00	3,00	3,00	5,00	2,25
Altibase	3,00	1,00	1,00	4,00	5,00	2,60
Oracle Event Processing	5,00	3,00	4,00	5,00	5,00	4,25
Drools	5,00	4,50	5,00	5,00	5,00	4,85

Table 2.4. Selection of the complex event processing tool.

As noted in the previous evaluation in accordance with the obtained results the complex event processing tools for the detection of risk situations and abnormal events within early warning systems with the best results are Esper, Oracle Event Processing, and Drools. It should be noted that although the final results of these tools were above 4 (very good), the most influential criterion in the final selection was *Open Source*, therefore Oracle Event Processing is discarded since it does not allow its complete installation and implementation for free, i.e. although some components may be used without any cost, several of its features are reserved only for developers who have acquired a license, unlike Esper and Drools which enable a full implementation of all their features for free.

In order to finally select a CEP tool other sources were consulted looking for the differences between Drools and Esper: [56] noted that Esper in comparison with Drools Fusion uses SQL-like syntax, which can be easier for programmers to learn, contains fewer temporal operators and may have better performance results within a short period of time; and [57] showed that Esper has less memory and CPU usage than Drools with almost the same performance. Hence, according to the previous evaluation and the external sources [56] [57], the selected complex event processing tool for the detection of risk situations and abnormal events within early warning systems is Esper.

Summary

This chapter presented the concepts and technologies related within this undergraduate thesis such as the JAIN SLEE specification, OpenBTS, complex event processing and early warning systems. Furthermore an analysis of the related works was performed dividing them in three groups: Deployment of warnings in converged environments, providing a set of different systems aimed at the dissemination of warnings through converged services in various scopes; Dissemination of converged services via OpenBTS, describing various environments where the OpenBTS network was implemented; and Complex event processing in converged environments, analysing various projects where the converged environments leveraged the features of complex event processing in order to improve the users' experience. This analysis concluded that, though there are very complete works, none of them achieves the integration of CEP and converged environments in a low-cost telecommunications network for the deployment of warnings in rural areas.

Finally a characterization and selection process of the tools and technologies was performed, starting with the selection process for the telecommunications service dissemination tool, where OpenBTS was selected; following with the selection process for the converged service development and deployment platform, where

Mobicents JAIN SLEE was selected; concluding with the selection process for the complex event processing tool, where Esper was selected.

Chapter 3

Improving early warning systems with CEP

This chapter presents a detailed description of the application scenario for this undergraduate thesis, considering a specific case study focused on the coffee rust early detection and warning. Besides, the definition of the different event patterns implemented within the experimental prototype is presented. Finally the integration architecture of CEP with converged environments is exposed, proposing the implementation of three of the four components of an early warning system: Risk Knowledge, Monitoring & Warning Service and Dissemination & Communication [18].

3.1. Case study

Coffee production is the main Colombian agricultural activity, more than 350.000 families depend on the coffee harvest for their sole income. Therefore, diseases, pests and even low prices have a negative impact on the economic and social aspects of the main coffee-growing regions. Coffee rust, first reported in 1983 [58], is the most important and severe disease currently affecting the production of Colombian coffee. Resistant varieties have been developed through DNA improvement with genes of Timor Hybrid (a plant with natural resistance to the disease) as a solution to the rust problem, yet more than 50 percent of the country's coffee crop is still susceptible in the productive phase. Studies on coffee rust have concluded that the spores carrying the infection are spread by climatic elements such as temperature, humidity, wind and rainfall, wind being the vector for long distance spore transport, while precipitation droplets are responsible for vertical propagation from infected leaves or soil. Once spores make contact with a susceptible leaf, the infection process is improved by high shadow index, high humidity (atmosphere and leaf), soil acidity, high coffee tree density and low soil fertility.

Since coffee rust has led to considerable losses in the industry worldwide, recent researches have focused on detecting the coffee rust disease. Colombian researchers [59] made use of meteorological variables (relative humidity average in the last 2 months, temperature variation average in the last month, accumulated precipitation in the last 2 months), physic crop properties (coffee variety, crop age, percentage of shade), and crop management (coffee rust control in the last month, fertilization in the last 4 months) with the aim of reporting the incidence of rust. In the same way, Brazilian researchers [60] [61] sought to notify the appearance of coffee rust considering meteorological variables (daily average of night hours with air relative humidity, wind speed sum average, average daily temperature), and physic crop properties (fruit load of the plantation, spacing between plants).

Despite the effort made by Colombian and Brazilian researchers to detect and notify the coffee rust, most of the existing rust detection and notification systems lack the ability to process information coming from multiple sources in real time, therefore an insufficient amount of context information may cause a significant delay or a mislead on the system reports. In spite of these disadvantages, these kind of systems can be improved by the implementation of two specific technologies: Telco 2.0, which allows to deploy improved reports through Web 2.0 and telecommunications services, and the Complex Event Processing (CEP), which allows to perform a real time analysis of multiple source information; hence taking advantage of the converged environments features and the CEP it is possible to implement an early warning system capable of detecting any abnormal situation to subsequently warn about its risks.

3.2. Event Patterns

This section presents the event patterns definition for the Risk Knowledge element of an early warning system focused on the detection and notification of the coffee rust disease appearance probability; hence the identification of risky situations and abnormal situations is described as follows.

3.2.1. Detection of risky situations

A risky situation is a serious, unexpected and dangerous occurrence which require an immediate intervention [62]; these threatening events may hold one or various types of hazards for the society, environment and economy, like natural disasters (wildfires, eathquakes, tornados, tsunamies, etc.) or man-made emergencies (explosions, radiological emergencies, chemichal threats, etc.) [63].

The objective of the emergency managers is to decrease the negative impact of risky situations, but as they usually react after the happening of this kind of events, they are unable to opportunely intervene in the emergencies, delaying the mitigation of the emergency-caused adverse consequences. Therefore a system that enables the prevention and early detection of risky situations represent an important advantage in the emergency management, since by establishing a constant surveillance and preparation before an specific risky situation the emergency managers can opportunely decrease the negative impact of such situation.

Most of the risky situations have specific characteristics which enable their identification through the constant monitoring of a set of parameters arranged as patterns; such is the case in the appearance of the coffee rust in colombian crops.

Detection of the coffee rust disease appearance probability

Considering that the coffee rust, caused by *Hemileia vastatrix*, is one of the most important diseases and pests of tropical plants, and the major cause of losses in the

Colombian coffee crops, there have been several projects aiming to predict the coffee rust appearance by implementing empirical and fundamental methods [64]. The empirical predictive systems are developed by analysing historical data records of disease occurrence and concurrent climate conditions in the same locality, without considering the pathway of biological action; while the fundamental predictive systems are developed from experimentally obtained data in a controlled environment, observing the relationships of the biological and environmental conditions governing the host-pathogen interactions [64]. Although the plant disease epidemiology can offer only empirical generalizations these models have been improved with more accurate principles obtained by continued experimentation, reducing the gaps between empirical and fundamental methods [64].

These projects have provided a large quantity of information on the coffee rust disease, observing that the fungus *Hemileia vastatrix* needs very specific conditions to parasitize the leaves of the coffee plant. Particularly, it requires rain splashes to begin the process of dispersion between leaves and later between plants, as well as the presence of a layer of water on the underside of the leaves to begin its germination, along with temperatures between 16 and 28 Celsius degrees and low sunshine conditions. Therefore, environments with constant rainfall, especially in the afternoon or evening, with occurrence of cloudy skies preventing a temperature rise in noon, or with very low temperatures in the early morning, are conducive to the development of strong coffee rust epidemics [65].

The most popular areas for coffee production in Colombia are located in the optimum range of the disease development, with an average temperature of 22 Celsius degrees. Until 2007, coffee plantations exposed at sun and established 1,600 meters above sea level, which corresponded to an average temperature of 19 Celsius degrees, did not require disease control methods; however, changes occurred since 2008 have generated favorable environments for the coffee rust development at multiple locations of the country. For example, in Colombia’s central coffee zone, during 2008 and 2010, rainfall records have shown increases up to 40%, the solar brightness has decreased and the relative humidity was over the 85% during several months, all of this favors the development of the disease. In September of 2008 the amount of rain was constant but its distribution was not even, and the days had hours of rain combined with high temperatures. Under these conditions, the rust was favored to the extent that at the end of the harvest period, the crops had reduced foliage and showed symptoms of incidence and severity [65].

Overall, the decrease in the difference between maximum and minimum daily temperatures, due to the occurrence of the weather phenomenon known as “La Niña” at all altitudes of the coffee region, and the constant presence of leaf moisture, caused that the coffee rust disease presented high levels with epidemic proportions in some areas above 1,600 meters of altitude, provoking die-back [66] even at 2000 meters, as occurred in the departments of Antioquia, Huila, Caldas and Quindio. Therefore, today it is recommended that the monitoring and control of the coffee rust disease must be rigorous in any altitude where coffee can be produced [65].

Such changes in the meteorological conditions, propitious for the coffee rust infection, have encouraged researchers to conduct deeper studies on the detection of the coffee rust disease appearance probability, concluding that the spores carrying the infection are spread by climatic elements such as wind and rainfall. Wind being the vector for long distance spore transport, while precipitation droplets are responsible for vertical propagation from infected leaves or soil. Once spores make contact with a susceptible leaf, the infection process is increased by high shadow index, high humidity (atmosphere and leaf), soil acidity, high coffee tree density and low soil fertility [58].

Brazilian machine learning researchers have built different datasets and decision trees in order to help the decision making when monitoring the coffee rust appearance probability [67] [60] [61], and following their example a Colombian approach proposed a new dataset based on information collected every three months (from 2011 to 2013) at the Experimental Farm *Los Naranjos* of Supracafé, in Cajibio, Cauca, Colombia [59]. This research defined 21 attributes divided in 4 categories: Weather conditions (6 attributes), Soil fertility properties (5 attributes), Physic crop properties (6 attributes), and Crop management (4 attributes) [59].

Based on obtained results of the aforementioned research, the selected attributes to be analysed in order to opportunely detect the coffee rust appearance probability are Relative Humidity, Temperature, Wind Speed and Pluviosity. Therefore, aiming to complete the Risk Knowledge element from an early warning system developed within this undergraduate thesis, a set of event patterns is defined and divided according to their corresponding coffee rust appearance probability [59].

The table 3.1 illustrates the defined event patterns for the early detection of the coffee rust disease in Colombian crops, the patterns are organized in three risk levels (Low, Medium and High); the Relative Humidity (Humidity) is measured in Percentage, the Wind Speed in meters per second, the Temperature in Celsius degrees and the Pluviosity in millimeters and the reference values correspond to a monthly average except for the Pluviosity, which is the monthly aggregate.

Alert Level	Coffee Rust Appearance Probability (%)	Event Patterns
1 - Low	3,65	91.087975 < Humidity <= 91.85424 AND Wind Speed > 0.864314 AND Pluviosity <= 97.79952 OR Humidity > 91.85424 AND Wind Speed > 0.864314 AND Temperature <= 16.60662
	4,32	Humidity <= 73.437775 AND 18.134516 < Temperature <= 18.479593 AND 14.800012 < Pluviosity <= 414.90765 OR Humidity <= 73.437775 AND 1.4 < Pluviosity <= 414.90765 AND 18.55548 < Temperature <= 19.166304 OR Humidity > 73.437775 AND Temperature <= 19.166304 AND Pluviosity <= 22.800016 AND Wind Speed > 1.06829
	4,93	86.14845 < Humidity <= 91.087975 AND Wind Speed > 1.017179 OR 86.14845 < Humidity <= 91.087975 AND Pluviosity <= 74.699684 AND 0.864314 < Wind Speed <= 1.017179

Table 3.1a. Low risk event patterns.

Alert Level	Coffee Rust Appearance Probability (%)	Event Patterns
2 - Medium	5,33	91.087975 < Humidity <= 91.85424 AND Wind Speed > 0.864314 AND Pluviosity > 97.79952 OR 73.437775 < Humidity <= 84.87477 AND Temperature <= 19.166304 AND Pluviosity <= 22.800016 AND Wind Speed <= 1.06829 OR 86.14845 < Humidity <= 91.087975 AND 0.864314 < Wind Speed <= 1.017179 AND Pluviosity > 74.699684 OR 91.85424 <= Humidity < 92.1369 AND 0.864314 < Wind Speed <= 1.020617 AND Temperature > 16.60662 AND Pluviosity > 134.09969 OR Humidity > 92.1369 AND Pluviosity > 134.09969 AND Temperature > 16.60662 AND Wind Speed > 0.864314
	5,99	Humidity > 84.87477 AND Wind Speed <= 0.864314 AND Temperature <= 17.465523 OR Humidity > 91.85424 AND Wind Speed > 0.864314 AND Temperature > 17.31166 AND Pluviosity <= 134.09969
	6,16	Humidity <= 73.437775 AND Pluviosity <= 1.4 AND Temperature <= 19.166304 AND Wind Speed > 1.401474 OR Humidity <= 73.437775 AND 1.4 < Pluviosity <= 414.90765 AND Temperature <= 18.134516 OR Humidity <= 73.437775 AND 1.4 < Pluviosity <= 14.800012 AND 18.134516 < Temperature <= 18.479593 OR Humidity <= 45.42533 AND Temperature > 19.166304
	6,22	Humidity > 84.87477 AND Temperature > 17.465523 AND Wind Speed <= 0.864314 OR Humidity > 91.85424 AND 16.60662 < Temperature <= 17.31166 AND Pluviosity <= 134.09969 AND Wind Speed > 0.864314 OR Wind Speed > 1.020617 AND 91.85424 < Humidity <= 92.1369 AND Pluviosity > 134.09969 AND Temperature > 16.60662

Table 3.1b. Medium risk event patterns.

Alert Level	Coffee Rust Appearance Probability (%)	Event Patterns
3 - High	8,9	Humidity <= 73.437775 AND 18.479593 < Temperature <= 18.55548 AND 1.4 < pluviosidad <= 414.90765 OR 73.437775 < Humidity <= 84.87477 AND Pluviosity > 22.800016 AND Temperature <= 19.166304 OR 59.482613 < Humidity <= 84.87477 AND Temperature > 19.163304 OR 84.87477 < Humidity <= 86.14845 AND Wind Speed > 0.864314
	10,47	Humidity <= 73.437775 AND Temperature <= 19.166304 AND Pluviosity <= 1.4 AND Wind Speed <= 1.401474 OR Humidity <= 73.437775 AND Pluviosity > 414.90765 AND Temperature <= 19.166304
	12,41	45.42533 < Humidity <= 59.482613 AND Temperature > 19.166304

Table 3.1c. High risk event patterns

3.2.2. Detection of abnormal situations

An abnormal situation is a critical, complex, unplanned and ambiguous event which compromises the proper expected performance of any kind of system. These events usually are all kinds of unexpected situations that leads to the malfunctioning (human, hardware, software, etc.) of a system. An early warning system has as objective the continuous monitoring of an specific kind of emergency and once a possible risky situation is detected, the system must proceed to warn the population about the emergency, hence if an abnormal situation provokes a malfunctioning of the early warning system it could lead to the generation of false alarms, obtain mistaken information about the monitored phenomenon, or even provoke a total absence of the reports the early warning system must be providing.

The abnormal situations present some characteristics that can be identified by analysing and comparing the expected and actual performance of the system; therefore, most of the events that indicate the malfunctioning of a system are intrinsic to itself. Hence, in order to detect abnormal situations on the implementation, the performance of the meteorological station must be considered.

Under normal circumstances a meteorological station is powered by the electrical network and has an Uninterruptible Power Supply (UPS) in case of a power failure. The station collects climate information from the sensors network about the four parameters previously described (relative humidity, temperature, wind speed and pluviosity), then every fifteen minutes the collected information is sent via HTTP; the provided information is the average of the relative humidity, the average of the temperature, the average of the wind speed and the accumulated pluviosity along the day. Therefore the expected failures of the meteorological station are:

- Sensors or Datalogger damage: the meteorological station presents an irregular behavior sending mistaken information like out of range data, drastic changes or illogical changes between readings, and repeated information for an extended period of time. Such behavior can be provoked by damages on the sensors i.e. the sensors take mistaken readings, or damages in the datalogger i.e. although the sensors gather correct readings, such readings are stored in a wrong way.
- Transmission system damage: The transmission system of the meteorological station fails presenting two possible conditions: when the station does not transmits any data or transmits only null values.
- Power failure: When the electrical network presents a blackout and the UPS runs out of energy the station stops transmitting information.

The Monitoring & Warning Service element from an early warning system developed within this undergraduate thesis leverages the climate information provided by several meteorological stations which periodically sends a set of data about the current environment of the coffee crops (pluviosity, temperature, wind speed, and relative humidity as humidity); hence it is necessary to take into consideration the different abnormal situations that may affect the system: Null Values, the received data only have null or zero values; Sensor Damage, when the incoming data gets out from the theoretical bounds, has an illogical value, has a repeated value or has drastic changes; and Absence of Incoming Information, when the system does not receive any data. The table 3.2 illustrates a set of event patterns related to the defined abnormal situations, which may compromise the performance of the early warning system.

Meteorlogical Station Failures	Abnormal Situation	Event pattern
Sensor or Datalogger Damage	Relative humidity out of range	Humidity < 0 OR Humidity > 100
	Temperature out of range	Temperature < -5 OR Temperature > 45
	Wind Speed or Pluviosity out of range	Wind Speed < 0 OR Pluviosity < 0
	Pluviosity illogical changes	Current Pluviosity < Previous Pluviosity
	Repeated readings of Humidity, Temperature or Wind Speed during 5 hours	Average Deviation Humidity = 0 OR Average Deviation Temperature = 0 OR Average Deviation Wind Speed = 0
	Drastic changes in readings of Humidity, Temperature and Wind Speed	Current Humidity >> Previous Humidity OR Current Humidity << Previous Humidity OR Current Temperature >> Previous Temperature OR Current Temperature << Previous Temperature OR Current Wind Speed >> Previous Wind Speed OR Current Wind Speed << Previous Wind Speed
Transmission System Damage	Null values	Humidity = 0 AND Temperature = 0 AND Wind Speed = 0 AND Pluviosity = 0
Power Failure	Absence of incoming information during 2 hours	Number of received events = 0

Table 3.2. Abnormal situations event patterns.

3.3. CEP and converged environments

The architecture of a CEP and Telco 2.0 integration for coffee rust warning illustrates the implementation of two of the four components of an early warning system: Monitoring & Warning Service, via the real time analysis of climate information done by the CEP component; and Dissemination & Communication through the services implemented inside the converged environments.

The CEP capacity of processing and analysing multiple source information in real time permits the automatic warnings deployment. These features represent an important advantage within early warning systems since by automating the detection process the warnings triggered become more accurate and efficient. On the other hand 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 complexity.

It is worth mentioning that this proposal takes into account some of the most common problematics of the rural zones from Colombia, where the coffee production is the main agricultural activity. More than 350.000 Colombian families depend on coffee harvest for their sole income; however diseases like the coffee rust can cause a big impact in the coffee production triggering a significant reduction in these families quality of life. Studies on coffee rust have concluded that the spores carrying the infection are spread by climatic elements such as wind and rainfall, the wind being the vector for long distance spore transport, while precipitation droplets are responsible for vertical propagation from infected leaves or soil [59]. The probability of coffee rust appearance can be detected by analysing these climatic elements, hence our proposal seeks to process the data received through the meteorological station and the prediction module, such information is verified and once a probable appearance of the coffee rust is detected, triggers automatic warnings aimed at the farmers and the local authorities through converged services.

The Figure 3.1 shows the architecture of a CEP and Telco 2.0 integration for the coffee rust warning based on the example found in [68] and [12]; it is composed by three main modules: the converged control module, the CEP module and the Access Module, which will be described as follows along with the two external components aimed at acquiring the climate data and predict the appearance of coffee rust.

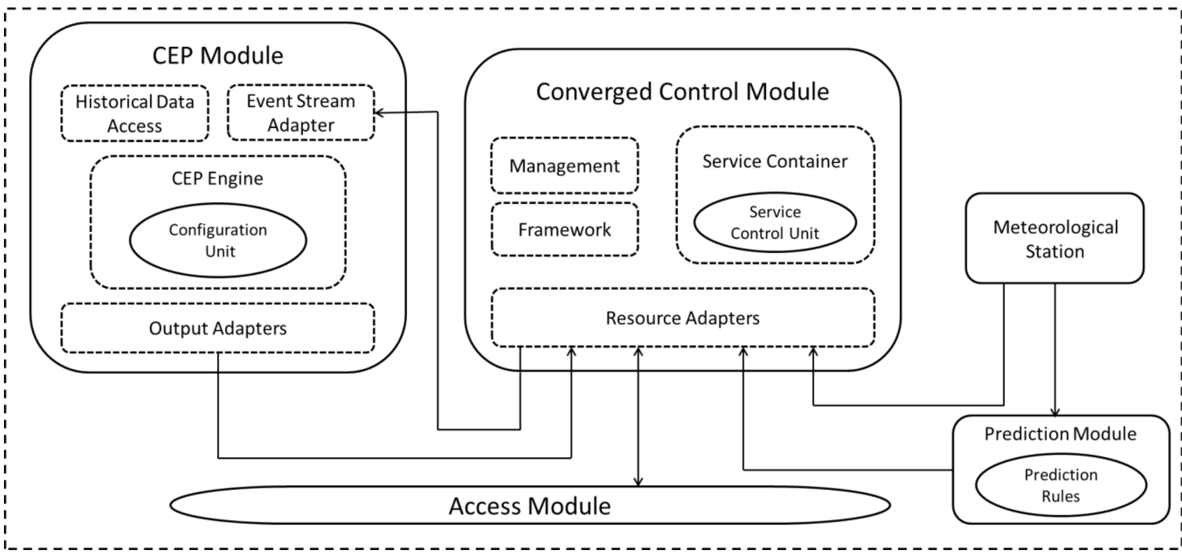

Figure 3.1. Architecture of a CEP and Telco 2.0 integration.

Meteorological station: The station collects climate information from the sensors network about the four parameters previously described (relative humidity, temperature, wind speed and pluviosity), then every fifteen minutes the collected information is sent via HTTP to the Prediction Module and the Converged Control Module.

Prediction module: This module receives the climate information from the meteorological station and analyses it in order to predict the appearance of coffee rust, when this module detects a real threat it sends a report to the Converged Control Module via HTTP.

Converged control module: This module represents the first part of the Dissemination & Communication element of an early warning system by handling the control and signaling operations needed by the converged services and it is composed by:

Service container: Manages all the logic related with the service behavior, like the resource access or the synchronous/asynchronous invocation. It is composed by the service control units, modular components which define how and when a service is executed i.e. what services must be executed according to the user requirements. A good example of a service control unit could be a service capable of posting a message on Twitter (Web 2.0 service) or a service capable of sending a SMS to a specific list of users (Telco service).

The Service Container executes three types of services: The Monitoring & Warning services related with the monitoring and warning of risky or abnormal situations, the Control & Support services related with the control and support operations needed inside the Converged Control module, and the Dissemination & Communication services related with the warnings dissemination and communication. All of these services will be explain in detail subsequently in sections 3.4 and 4.2.

Management component: This component specifies the mechanisms to manage the converged environment, as design patterns, various API and monitoring services.

Framework component: This component contains specific software modules, which are the basis for the organization, development and deployment of the services inside the converged environment.

Resource adapters: This component adapts the information exchanged through various communication protocols (HTTP, SIP, SMTP, SQL, etc.) by the converged environment and the external components to a suitable format for the platform.

This module receives the climate information from the Meteorological Station via HTTP, this data is passed to the CEP module to be analysed as a java object, aiming at detecting the coffee rust appearance probability and the occurrence of abnormal situations, on the other hand, it receives the threat alerts from the Prediction Module and, according to the previous analysis of the received climate information, proceeds to initiate the dissemination process for the detected warning if it is necessary.

CEP module: This module represents the Monitoring & Warning Service component of an early warning system by handling all the complex event processing operations, i.e. it performs a real time analysis of the incoming information to subsequently generate an automatic warning if an abnormal situation is detected. It is composed by:

> *CEP Engine:* This component processes all the incoming events and analyses them in real time according to the configuration parameters stored in the configuration unit, triggering different types of reports. The configuration parameters (analysis conditions, number of warnings per day, number of events received, etc.) are set by the Risk Knowledge component of an early warning system and define which warnings answer to each condition.
>
> *Event Stream Adapter:* This adapter communicates with the converged environment and receives all the events from the platform so they can be processed by the CEP engine, the received events are represented as java objects containing the collected climate information from the Meteorological Station.
>
> *Historical Data Access:* This module stores the previous received events in order to help the CEP Engine to process and analyse the incoming events.
>
> *Output Adapters:* This module is an output interface for the CEP Engine, formatting its reports or warnings into standardized results, to finally return such reports to the Converged Control Module via HTTP.

Access module: This module represents a part of the Dissemination & Communication element of an early warning system by referring to all the transport nodes within the integration architecture and acting as an interface which allows accessing the system functionalities.

3.4 Service catalog

A service catalog is an organized collection of all business and technology information related with the services that can be offered by any Service Provider [69] [70]. Service catalogs act as knowledge management tools for the employees, consultants and customers of an enterprise, allowing them to comprehend and lead their requests for any service stored in the catalog that can satisfy their specific requirements. Each service within the catalogs typically includes elements such as: ownership, identification, description, categorization, request types, supporting services, among others [70].

Therefore, as service catalogs are a very useful tool for describing, understanding and managing services, the service catalog for the implemented platform within this undergraduate thesis, aiming to opportunely detect and notify risky or abnormal situations in Colombian coffee crops, is divided in three categories: Monitoring & Warning services, Dissemination & Communication services, and Control & Support services. This section describes the services related with the improving of early warning systems with CEP, classified as Monitoring & Warning services.

3.4.1. Monitoring & Warning services

This subsection presents the surveillance service which is composed by the Service Building Blocks (SBB) related with the monitoring and warning of risky or abnormal situation. This service leverages the information provided by the Meteorological Station and the Prediction Module besides the functionalities of the Access module, the CEP module and the Converged Control module of the platform implemented within this undergraduate thesis illustrated in the Figure 3.1.

Surveillance service

This service receives the climate information from the Meteorological Station via HTTP, subsequently processes and analyses the received data in order to determine the warnings level and type, furthermore receives coffee rust appearance forecasts from the Prediction Module and finally the results are sent to the next service. The SBB composing this service are:

Event receiver SBB

The event receiver SBB handles all the web communication processes with meteorological stations which provide the data related with the four climate parameters defined in section 3.2, also, it receives forecasts from the Prediction Module which will be validated according to the results obtained through the CEP SBB. Due to the stations' configuration, every fifteen minutes this SBB receives three HTTP requests carrying data about the relative humidity, temperature, wind speed and pluviosity of the monitored coffee crops, to subsequently send such information to the CEP SBB so it can be evaluated.

CEP SBB

The Complex Event Processing SBB receives the climate information from the event receiver SBB, thereafter processes such information through the event patterns defined in section 3.2 and illustrated in tables 3.1 and 3.2. Once the processing is finished, it can reach two different results: coffee rust appearance probability detected, if the climate data fits into the event patterns defined for the risky situations; or system malfunctioning detected, if the received information fits into the event patterns defined for the abnormal situations. Each analysed result is delivered to the threat alerter SBB, along with its corresponding error code or appearance probability.

Threat alerter SBB

The Threat alerter SBB receives the analysed results from the CEP SBB; such results allow this SBB to reach seven different conclusions depending of the identified situation and the validated forecasts from the Prediction Module:

Risky Situations: deploy Low Risk warnings, if there is a coffee rust appearance probability between 3% and 5%; deploy Medium Risk warnings, if there is a coffee rust appearance probability between 5% and 7%; or deploy High Risk warnings, if there is a coffee rust appearance probability superior to 7%.

Abnormal situation: alert about Null Values, if the received data only have null or zero values; alert about Data Out of Range, if the incoming data gets out from the theoretical bounds; alert about Sensor Damage, if the incoming data gets out from the theoretical bounds, has an illogical value, has a repeated value or has drastic changes; or alert about Absence of Incoming Information, when the platform does not receive any data.

Finally, this SBB proceeds to activate the orchestrator service called Service Shooter SBB which will activate none, some or all the Dissemination & Communication services, according to the reached conclusion.

Summary

This chapter presented a detailed description of the application scenario for this undergraduate thesis, considering a specific case study focused on the coffee rust early detection and warning, aiming to opportunely decrease the negative impact this disease has on the coffee farmers' production and quality of life. Besides, based on previous investigations, four climatic attributes (Relative Humidity, Temperature, Wind Speed and Pluviosity) were considered in order to define the different event patterns to detect the coffee rust appearance probability and the malfunctioning of the system (risky situations an abnormal situations), completing the implementation of the Risk Knowledge element of an early warning system.

Finally the integration architecture of CEP with converged environments was presented, illustrating the implementation of two of the four components of an early warning system: the Monitoring & Warning Service element, represented by the CEP Module; and the Dissemination & Communication element, partially illustrated by the Converged Control Module and the Access Module.

Chapter 4

Deploying converged services on rural environments

This chapter presents a detailed description of the integration between OpenBTS within converged environments aimed at rural zones of developing countries and finishes the implementation of the Dissemination & Communication module of an early warning system developed within this undergraduate thesis. Furthermore the service catalog implemented within the platform is explained and finally an application example of the system is illustrated.

4.1. OpenBTS and converged environments

Deploying warnings via converged environments in rural areas of developing countries, like Colombian coffee plantations, is not an easy task since a NGN is needed for the converged services deployment and this kind of networks are specially oriented to cities with a large population; besides the lack of advanced technology and investment resources caused by the absence of industry enterprises which do not find a profitable outcome in investing in these areas make the implementation of all kinds of projects a much more difficult task.

However, 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 the coffee rust, by watching over a set of climatic parameters, and once a risky or abnormal situation is detected (Section 3.2) it proceeds to warn the farmers and the people involved with the plantation through the implemented converged services. The Figure 4.1 shows the integration architecture based on the example found in [12], which is complementary to the CEP and Telco 2.0 integration architecture illustrated in the Figure 3.1, found in Section 3.3.

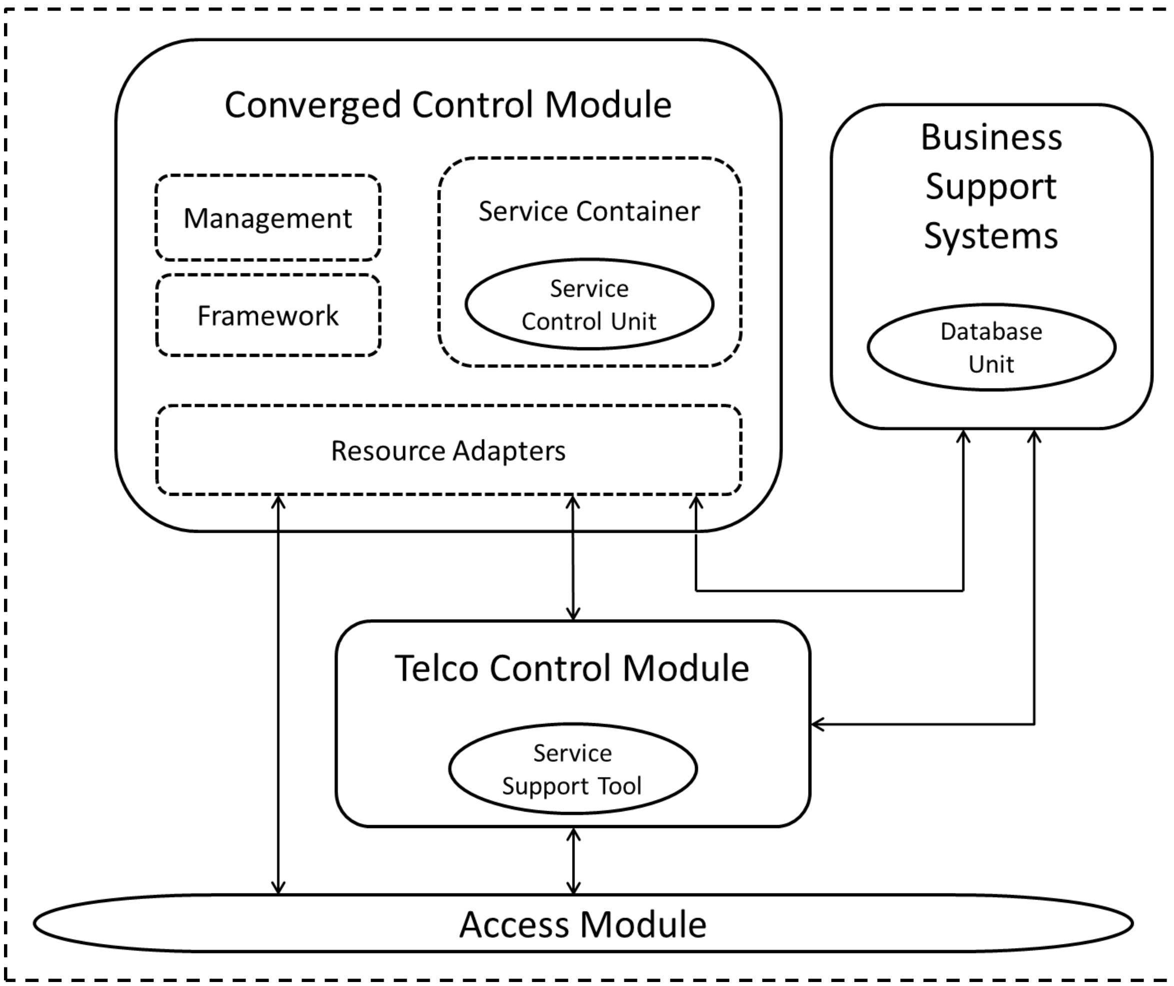

Figure 4.1. OpenBTS and Converged Environments integration.

The Figure 4.1 is composed by four modules, as the Converged Control Module and the Access Module were explained in Section 3.3, the Telco Control Module and the Business Support Systems will be explained as follows:

Telco control module: This module represents the second part of the Dissemination & Communication element of an early warning system by handling the control and signaling operations (communication establishment, switching, ending, etc.) needed to run the telecommunications services; therefore the Telco Control Module complements the converge control module by enabling the offering of mobile telephony and mobile data services.

This module is composed by the service support tools which are software components that allow it to manage the signaling, control and deployment of traditional Telco services like voice call or SMS. It is worth mentioning that the service support tools sustain the deployment of telecommunications services, while the service control units from the Converged Control Module manage all the logic related with the converged services behavior, like the resource access or the synchronous/asynchronous invocation. Besides, the communication between the Telco Control Module and the Converged Control Module is performed via SIP through the resource adapters.

Business support systems: The support system represents the third part of the Dissemination & Communication element of an early warning system by fulfilling all the features necessary to store the subscribers' information (farmers and local authorities' personal information like phone number and email address) and configuration parameters of the network.

The subscribers are divided in four different categories according to the their responsibilities within the coffee plantations: the farmers receives all the coffee rust appearance warnings generated by the system, the farm managers receive the medium and high risk warnings, the local authorities receive the high risk warnings and finally the technical personnel receive the damages and abnormalities warnings. Considering that this modules contains databases only, it communicates with the Converged Control Module and the Telco Control Module through SQL.

The Converged Control Module stores and executes three types of services: The Monitoring & Warning services, the Control & Support services and the Dissemination & Communication services. These services will be explained in detail within the service catalog presented in the next section.

4.2. Service catalog

This section continues with the service catalog presented in section 3.4, explaining the services that enable and support the deployment of warnings, classified as Control & Support services and Dissemination & Communication services.

4.2.1. Control & Support services

This subsection presents the control service, related with the control and support operations needed inside the Converged Control module, leveraging the Business Support Systems of the platform implemented within this undergraduate thesis.

Control service

This service receives the warnings information from the surveillance service and subsequently activates the Dissemination & Communication services according to each warning level and type. The control service is composed by the following SBB:

Service shooter SBB

The service shooter SBB is intended to trigger the Short Message Service (SMS), the Voice Call Service, the Email Service and/or the Twitter Post Service, according to the information provided by the Threat alerter SBB. This SBB needs to access the subscribers databases through the database SBB in order to select the users who will receive the risky or abnormal situation warnings.

Database SBB

The Database Service Building Block communicates with the subscribers databases (allocated in the Business Support Systems) and allows other SBB to access all the information stored in them; this SBB is important for the service shooter SBB since it needs to access the subscribers' information.

4.2.2. Dissemination & Communication services

This section presents all the services related with the warnings dissemination and communication, which leverage the Converged Control module, the Telco Control module and the Access module of the platform implemented within this undergraduate thesis.

Voice Call Service

This service allows to send a prerecorded message to a set of selected users, it is composed by the following SBB:

Call SBB

The Call SBB performs all the necessary signaling operations for the establishment of voice calls implementing the SIP protocol, sending an invite message to a predefined callee and establishing a session between both end points. This service is supported by the features of the B2BUA SBB and the Softswitch.

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, supporting the deployment of the SIP-based services (SMS and voice call). This SBB needs to access the subscribers databases through the database SBB in order to successfully mediate the SIP signaling.

The signaling process for this service is illustrated in Figure 4.2.

Short Message Service

This service allows to send a predefined text message to a set of selected users, it is composed by the B2BUA SBB and the SMS SBB, explained as follows:

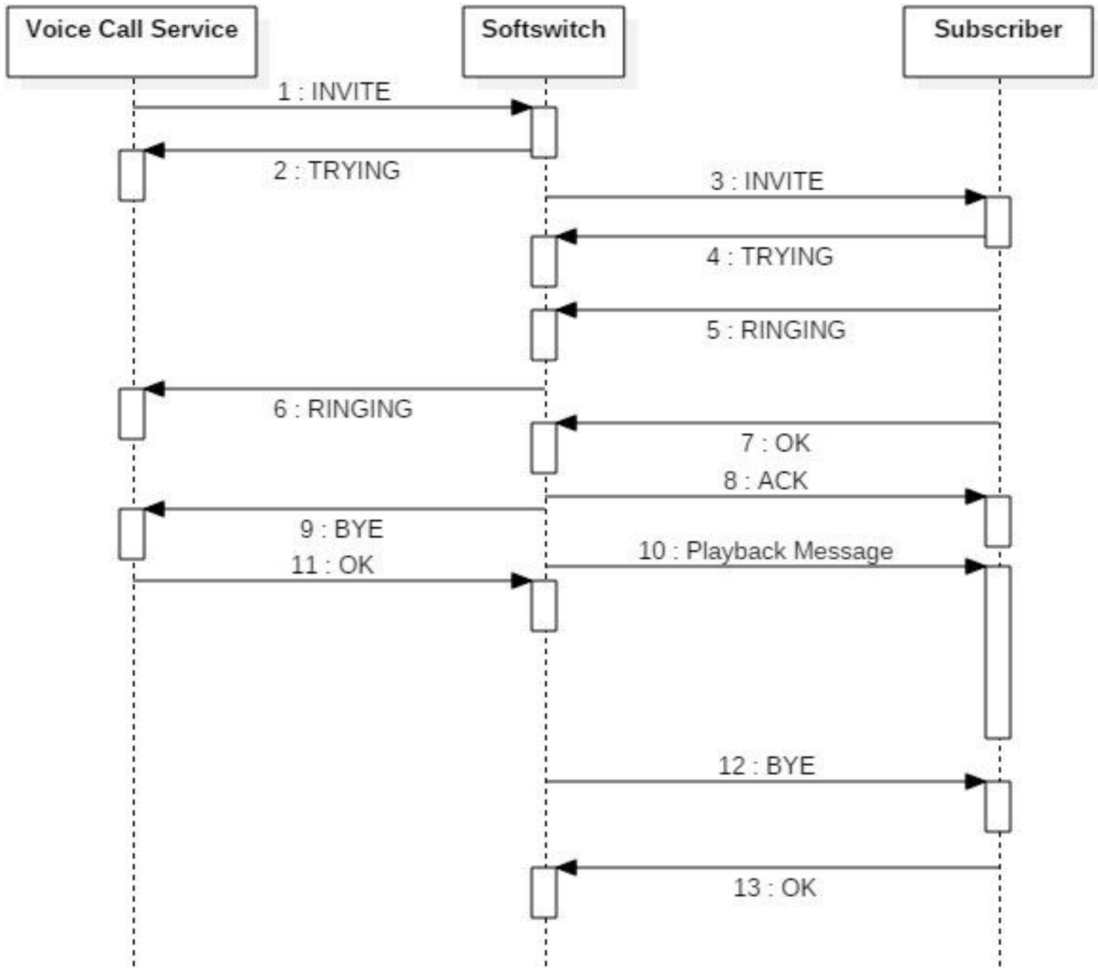

Figure 4.2. Voice Call Service SIP signaling.

SMS SBB

The Short Message Service SBB creates and sends a SIP instant message to a predefined user without establishing a session. This service is supported by the features of the B2BUA SBB.

Email Service

This service allows to send an email message containing specific warning data to a set of selected users, it is composed by the Email SBB, explained as follows:

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 Post Service

This service allows posting an update to Twitter with specific warning data, it is composed by the Twitter SBB explained as follows:

Twitter SBB

This SBB is capable of posting a warning message in an official Twitter account for the project through the Twitter API.

4.3. Application example

This section presents a brief application example to illustrate the operation of the early warning system described along previous chapters (3 and 4). It is worth remembering that the system is aimed at the detection of the coffee rust disease in rural areas of developing countries such as Colombia, considering that the damage provoked by the disease leads to a yield reduction around 35% in regions where the meteorological conditions are propitious for the disease development decreasing the farmers' quality of life, furthermore, the technological deficiency in the coverage of telecommunications services hinders the deployment of large scale projects leaving the population vulnerable before any kind of emergency situation.

The Figure 4.3a illustrates step by step the operation of the platform developed within this undergraduate thesis and the behavior of its external elements. The first stage of the external elements behavior relates the selected observation target (Coffee Plantations), due to its vulnerability to the selected risky situation (coffee rust appearance). The second stage, Meteorological Data Collection, represents the monitoring stations that collect information about the four climate parameters defined in section 3.2 (relative humidity, temperature, wind speed and pluviosity) from the observation target, with the aim of sending the collected climate data via HTTP to the platform developed within this undergraduate thesis and to the Coffee Rust Prediction System which analyses the received data according to a predefined dataset in order to generate forecasts which are also sent via HTTP to the developed platform.

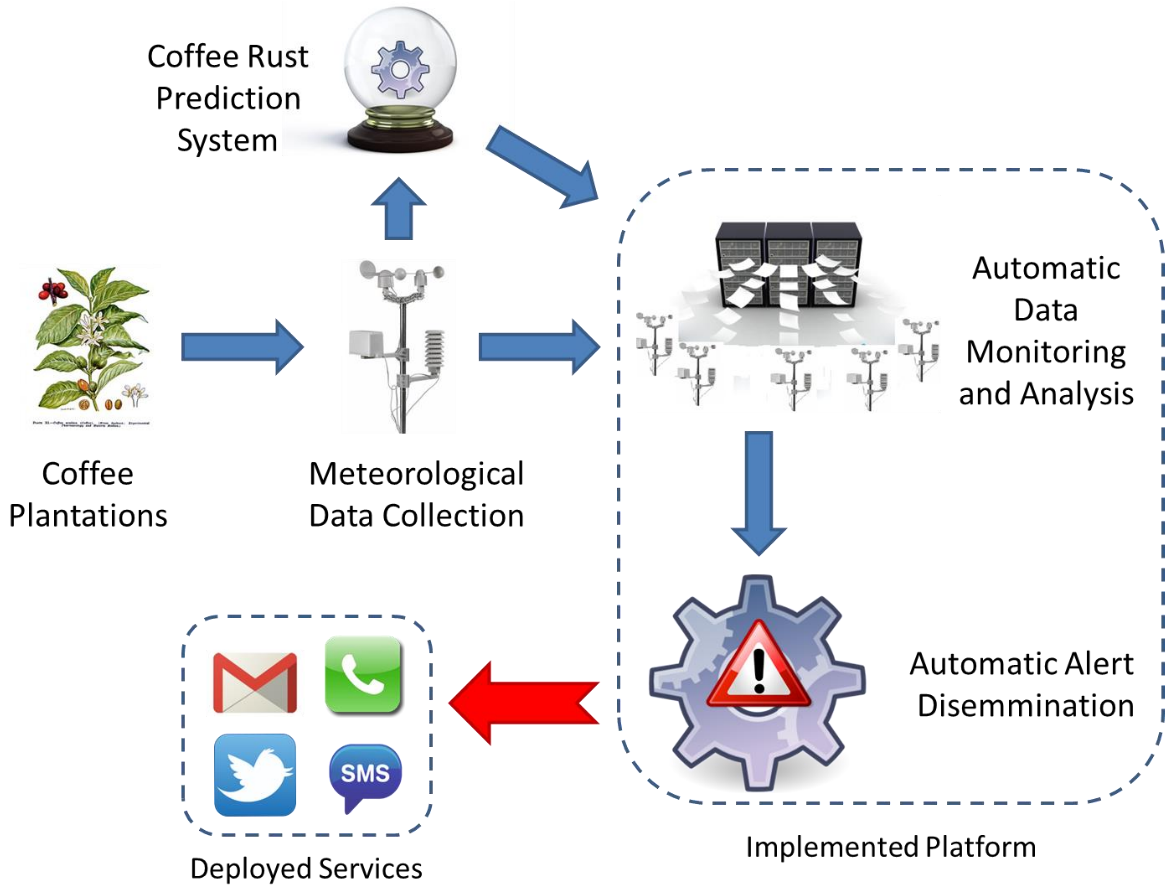

Figure 4.3a. Application example.

The implemented platform is represented by the blue box in the Figure 4.3a and its operation is illustrated in Figure 4.3b. The first stage in the platform's representation is the Automatic Data Monitoring and Analysis, which is represented by the Monitoring & Warning services and consists of three parts: data input (event receiver SBB), information analysis (CEP SBB), and warnings generation (thread alerter SBB). Once a forecast is received from the Coffee Rust Prediction System, the platform validates it as a risky or an abnormal situation by analysing the incoming information from the Meteorological Data Collection, subsequently the corresponding warnings are sent to the next stage, Automatic Alert Dissemination, which aims to automatically disseminate the received warnings through the Dissemination & Communication services (voice call, SMS, email, Twitter post), underpinned on the Control & Support services. If a low risk alert is validated the system proceeds to notify the population through the email and twitter post services; if a medium risk alert is validated the system activates the email, Twitter post and SMS services to proceed with the notification; in case of a high risk alert the system implements all the services to warn the population; and finally if an abnormal situation is identified the system notifies the system's technicians about the malfunctions through the email service. The abnormal situations that can be identified, explained in the subsection 3.2.2, are: Sensor or Datalogger Damage, Transmission System Damage and Power Failure.

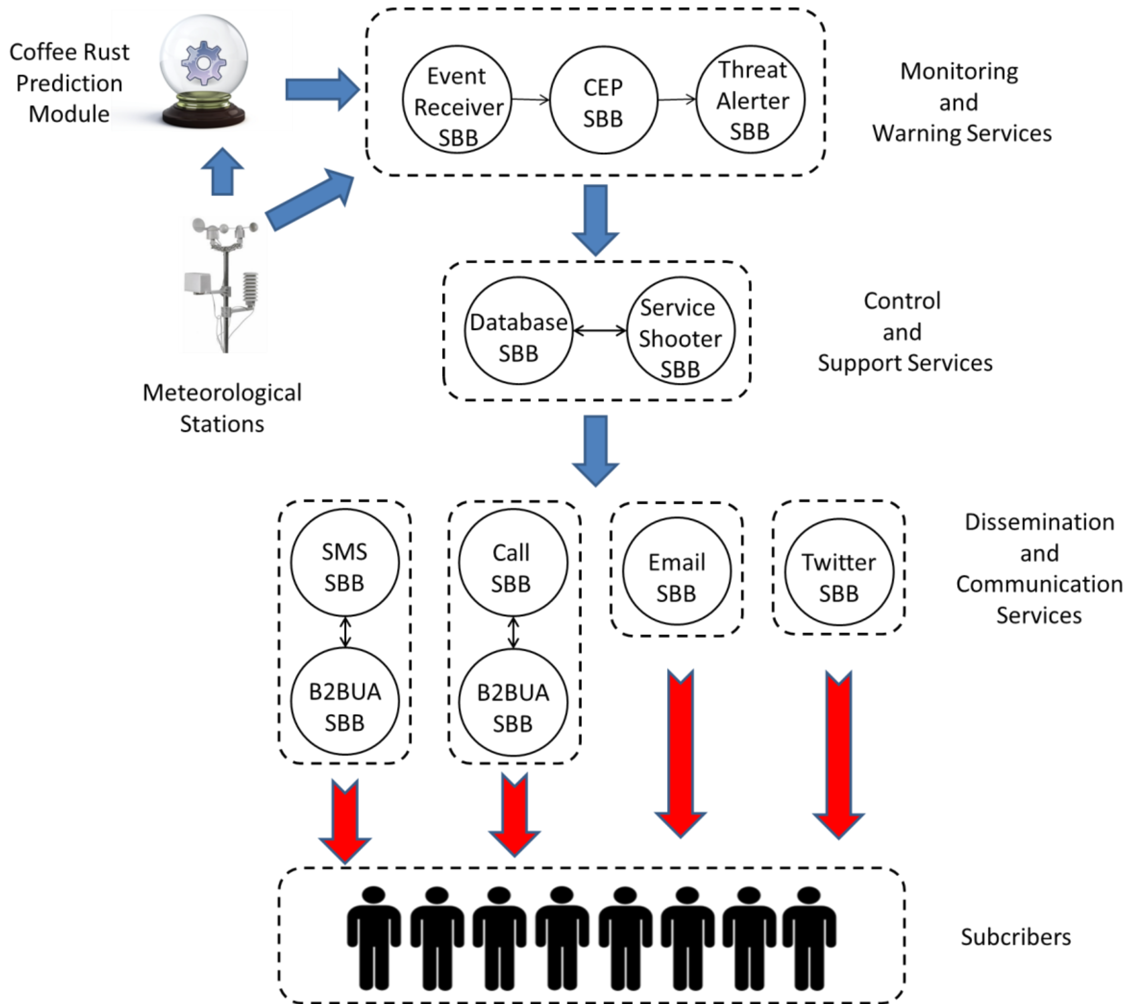

Figure 4.3b. Implemented Platform

Summary

This chapter presented a detailed description of the architecture aimed for the integration of OpenBTS within converged environments, enabling the platform developed within this undergraduate thesis to provide converged services in rural zones of developing countries, completing the implementation of the Dissemination & Communication module of an early warning system initiated in section 3.3. Furthermore the service catalog for the opportunely detection and notification of risky or abnormal situations relating the coffee rust within the implemented platform was explained, classifying the services in three categories: Monitoring & Warning services, Control & Support services and Dissemination & Communication services. Finally an application example focused on the coffee rust appearance probability detection and notification (section 3.1) was illustrated, explaining each stage of the platform's operation and its external components behavior.

Chapter 5

Experimental prototype

This chapter presents the experimental prototype of the platform implemented within this undergraduate thesis, with the aim of providing rural early warning systems with the capabilities of detecting and notifying risky or abnormal situations, concerning the coffee rust appearance in Colombian coffee crops. In the first place the deployment module of the experimental prototype is illustrated and its components are explained; in the second place the experimental prototype is tested in three different ways, presenting the tests' objectives, methodology and results.

5.1. Deployment model

In order to develop an experimental prototype following the architecture described in section 1.5 and observed in Figure 1.1, a set of specific technologies were selected in section 2.3 to handle the functionalities of the main components of the architecture and hence generating the deployment design, which is illustrated in the Figure 5.1 and described as follows:

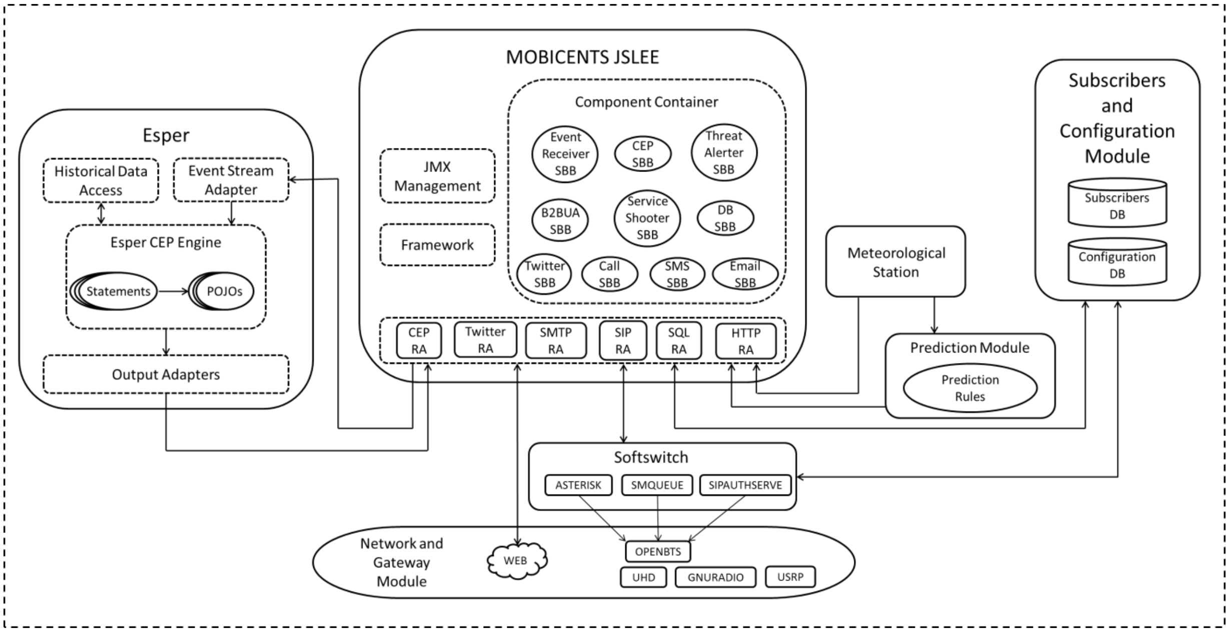

Figure 5.1. Deployment model.

Converged Control Module

Mobicents JAIN SLEE: The Mobicents Platform is the first and only open source VoIP platform certified for JAIN SLEE 1.1 and SIP Servlets 1.1 compliance. Mobicents serves as a high-performance core for Service Delivery Platforms (SDPs) and IP Multimedia Subsystems (IMSes) by leveraging J2EE to enable the convergence of data and video in Next-Generation Intelligent Network (NGIN) applications [71].

This platform is composed by the Service Container, the Resource Adapters, the JMX Management and the Framework components. The service container holds the Service Building Blocks (SBB), i.e. the service execution logic; the Resource Adapters (RA) provide the SLEE with the capabilities to connect with external components; the JMX Management allows the developer to control the environment; and the Framework includes all the SLEE facilities which can be leveraged by each SBB. The SBB needed for this prototype, fully explained in section 4.2, are briefly described as follows:

> *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 predefined 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 trigger the SMS SBB, the Call SBB, the Email SBB and/or the Twitter SBB with the rust information or the abnormal situation identified by the Threat Alerter SBB.

Event Receiver SBB: This SBB receives the HTTP requests with the climate data from the meteorological stations and the prediction module and subsequently routes the received information to the CEP SBB.

CEP SBB: This SBB receives the information from the Event Receiver SBB and sends it to the CEP Module (Esper) through the CEP RA; this SBB also receives rust incidence probability notifications from the CEP Module, which are sent to the Threat Alerter SBB.

Threat Alerter SBB: This SBB receives the notifications from the CEP SBB and the information sent by the prediction module through the Event Receiver SBB, sorting them in three different levels according to the rust incidence probability or in abnormal situations and finally activating the Service Shooter SBB.

The RA needed for this prototype are:

SIP RA: This Resource Adaptor provides the SLEE with the capabilities to handle all the SIP events.

SQL RA: This Resource Adaptor provides the SLEE with the capabilities to connect with SQL databases, mainly SQLite.

SMTP RA: This Resource Adaptor provides the SLEE with the capabilities to open a SMTP session with a preset email server.

Twitter RA: This Resource Adaptor provides the SLEE with the capabilities to access the Twitter functionalities through an API key.

CEP RA: This Resource Adaptor provides the SLEE with the capabilities to connect with the CEP Module through EPL (Event Processing Language).

CEP Module

This module handles all the complex event processing operations communicating with the converged control module and it is composed by:

Esper: Esper Event Processing capabilities features both Event Series and Complex Event Processing in one self-contained solution. Its unique capabilities for event series filtering, continuous queries, aggregation and joins, and its event pattern recognition capabilities ensure rapid implementation of the business-critical situation detection scenarios [68].

Event Stream Adapter: This component receives all the data streaming from the CEP SBB to pass it to the EPL statements defined in the Esper CEP Engine.

Esper CEP Engine: This component performs all the real time analysis to the climate information received through the event stream adapter based on the configuration statements and the previous information stored in the historical data access. Once an event is detected proceeds to send the rust incidence probability as a POJO (Plain Old Java Object) to the Output Adapter.

Historical Data Access: This module stores the previous received information in order to help the Esper Engine to process and analyze the incoming events.

Output Adapters: This module is an output interface for the Esper Engine, sending the rust incidence probability obtained from the EPL statements to the CEP SBB.

Telco Control Module

This module handles the control and signaling operations needed to run the telecommunications services and it is composed by:

Softswitch: 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 the voice calls by leveraging the SIP protocol.

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

- Sipautserve: This component handles all the SIP registration and authorization requests, it is also used to process the location update requests from OpenBTS and perform corresponding updates in the subscriber registry database [7].

Subscribers and Configuration Module

This module is implemented through relational databases [7], within them are two components:

Subscribers DB: This database contains all the information related to the users needed within the prototype.

Configuration DB: This database contains the entire configuration data needed within the Telco Control Module.

Access Module

This module refers to all the transport nodes from the OpenBTS infrastructure and the OpenBTS software features which handle the communication between the SIP and GSM protocols including user register, establishment of voice calls and instant messaging requested by the network users, also it includes the web access performed by the Twitter and Email service. It consists of five components which installation guide is provided in the Appendix A:

OpenBTS: This component handles all the incoming requests from the GSM endpoints, translates them into the SIP protocol and communicates with any of the previous components according to the GSM request received (Call, SMS, Registration) [7].

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 (OpenBTS) uses to control the USRP hardware and transmit and receive data.

UHD: The USRP Hardware Driver is the 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 the 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.

Web: This component enables the communication required by the Twitter and Email services through the IP protocol.

Meteorological station and Prediction module

The station collects climate information from the sensors network about the four parameters previously described (relative humidity, temperature, wind speed and pluviosity), then every fifteen minutes the collected information is sent via HTTP to the Prediction Module and the Converged Control Module. The Prediction Module receives the climate information from the meteorological station and analyses it in order to predict the appearance of coffee rust, when this module detects a real threat it sends a report to the Converged Control Module via HTTP. These components are external to the development of this undergraduate thesis and a complete description can be found in [59] [72].

In order to evaluate the experimental prototype the Meteorological Station and Prediction Module were simulated through Java applications as data sources for the platform. It is worth mentioning that the information sent by the simulation applications were obtained from a 5 year historical record of a real meteorological station surveilling coffee plantations [59].

5.2. Tests

This section presents the different tests that were conducted in order to observe the behavior of the experimental prototype, specifying the test objectives, the implemented methodology and illustrating the obtained results.

5.2.1. Test objectives

In order to evaluate the performance of the experimental prototype the following objectives were stated:

- Observe the Execution Time of the implemented dissemination and communication services, i.e. how much time each service needs to complete all its processing for each user.
- Observe the Success Rate of the implemented dissemination and communication services, i.e. how many users successfully received the service.
- Observe the latency of the complex event processing components, i.e. how much time the experimental prototype takes to finish the analysis of a predefined set of events.
- Observe the Success Rate of the complex event processing components through a Pollution test, where the percentage of successfully analyzed events is obtained, e.g. a set of 100 events are sent to the experimental prototype where 50 of those events are risky or abnormal situations, then the objective of this test is to observe how many of those 50 events are successfully detected.
- Perform a load test in order to observe the performance of the CEP components, measured in the percentage of CPU Load and Memory Usage facing an increasing stream of events.

5.2.2. Methodology

In order to evaluate the Execution Time and the Success Rate of the dissemination and communication services sixteen tests were conducted. Considering that this prototype is aimed for 8 users [7], four tests for every individual service were performed, repeating every test 10 times and dividing them according to the number of users (1 user, 5 users, 10 users and 15 users). Furthermore 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 10 times and is divided according to the number of users (1 user, 5 users, 10 users and 15 users).

Finally to evaluate the performance of the complex event processing components two main tests were conducted simulating 18 meteorological stations as the data sources and one prediction module through a Java application. The first test was the Latency and Pollution Test, the latency by observing how much time the prototype takes to finish the analysis of a predefined set of events measured in seconds (Latency), and the Pollution by observing the percentage of successfully analyzed events (Success Rate). The second test was the Load Test observing the performance of the server measured in the percentage of CPU Load and Memory Usage facing an increasing stream of events.

5.2.3. Results

The Figures 5.2a and 5.2b illustrate the results obtained in the tests performed to each service, the red line stands for the Success Rate in terms of percentage and the blue bars stand for the Execution Time measured in seconds and is divided according to the Users Number in each test performed.

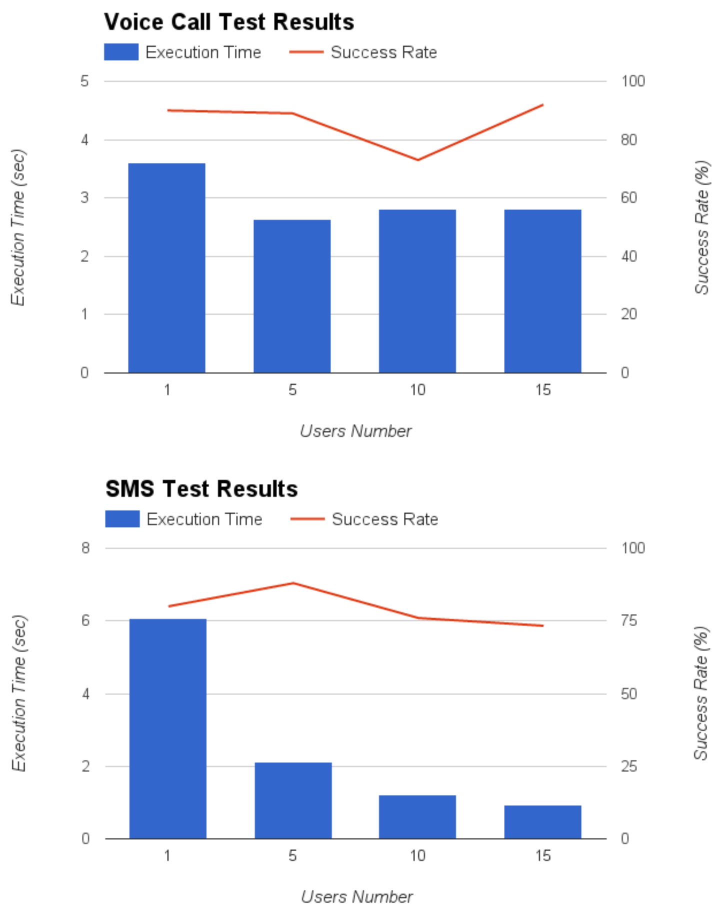

Figure 5.2a. Telco dissemination and communication services test results.

On the Figure 5.2a two graphics can be observed, the Voice Call results graphic shows an almost even Execution Time in every test and a considerable drop in the Success Rate for the test with 10 users, where the test experienced difficulties with the USRP which could not establish a communication with the phones within the GSM network showing a user unreachable exception. The SMS results graphic shows a decrease in the Execution Time when the Users Number increases, being one of the fastest services on the 15 users test, this happens because the service itself is very light in terms of processing, hence accessing the database is a critical stage in the Execution Time of the SMS, on the other hand the Success Rate shows an uniform performance in all the tests, however such result is barely acceptable and it is provoked by a compatibility problem between the SMS service stored in the service container and the native SMS service of OpenBTS; such inconvenient cannot be resolve easily due to the closed architecture of OpenBTS.

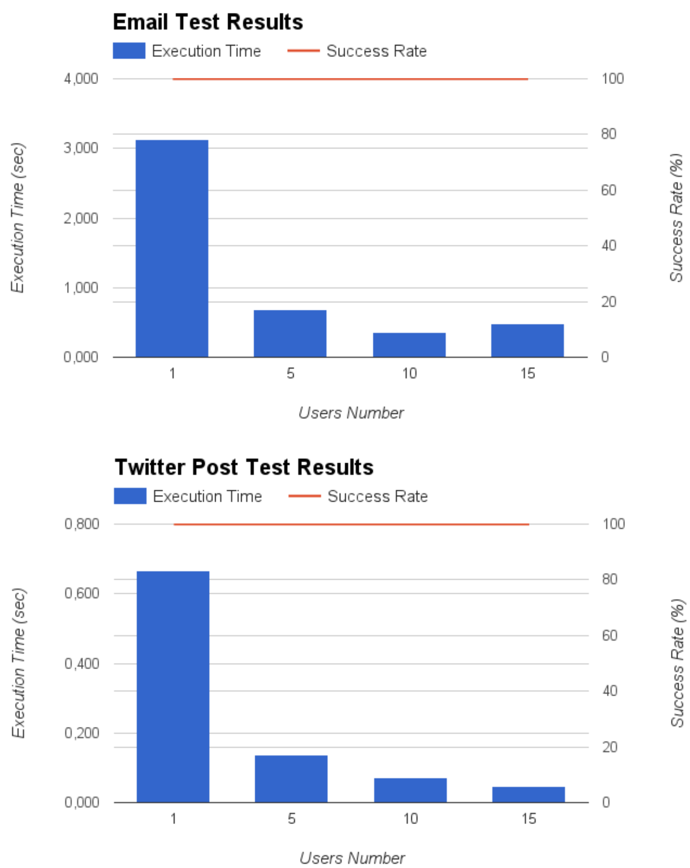

Figure 5.2b Web dissemination and communication services test results.

The Figure 5.2b shows two graphics, the Email results and the Twitter results graphics illustrate the best Success Rate of all the services obtaining a 100% score in every test, their Execution Time decreases when the Users Number increases, this happens because these services must establish a connection with the Gmail or Twitter servers, hence this connection causes an small delay but it hardly fails. It is worth mentioning that the Twitter service only access to its server once on every test, and the Email service must access its server once per every 10 users, therefore it makes one connection for the three first tests (1, 5 and 10 users) and two for the fourth test (15 users).

The results illustrated on Figures 5.2a and 5.2b evidenced that the Telco services (voice call and SMS) had the lowest (but still acceptable) Success Rate among all the services, dropping under the 75% in some tests, and the Web services had the best performance in both the Execution Time (under 3,1 seconds) and the Success Rate (100% on every test).

The Figure 5.3 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 Users Number in each test performed.

On the Figure 5.3 three graphics can be observed, considering three different alert levels. The 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. The 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. The Level 3 Alert Test results graphic has the same behavior in the Execution Time than the 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 and the protocols exchange (SIP to GSM) within the OpenBTS software. 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.

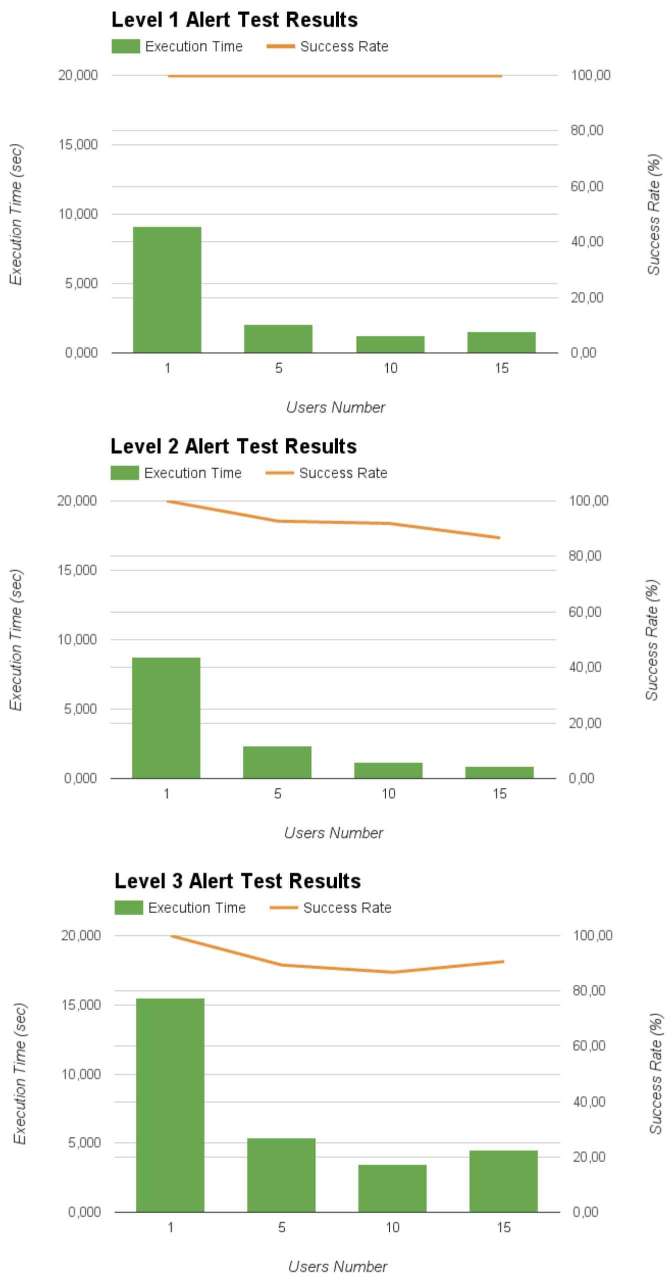

Figure 5.3. Alert levels test results.

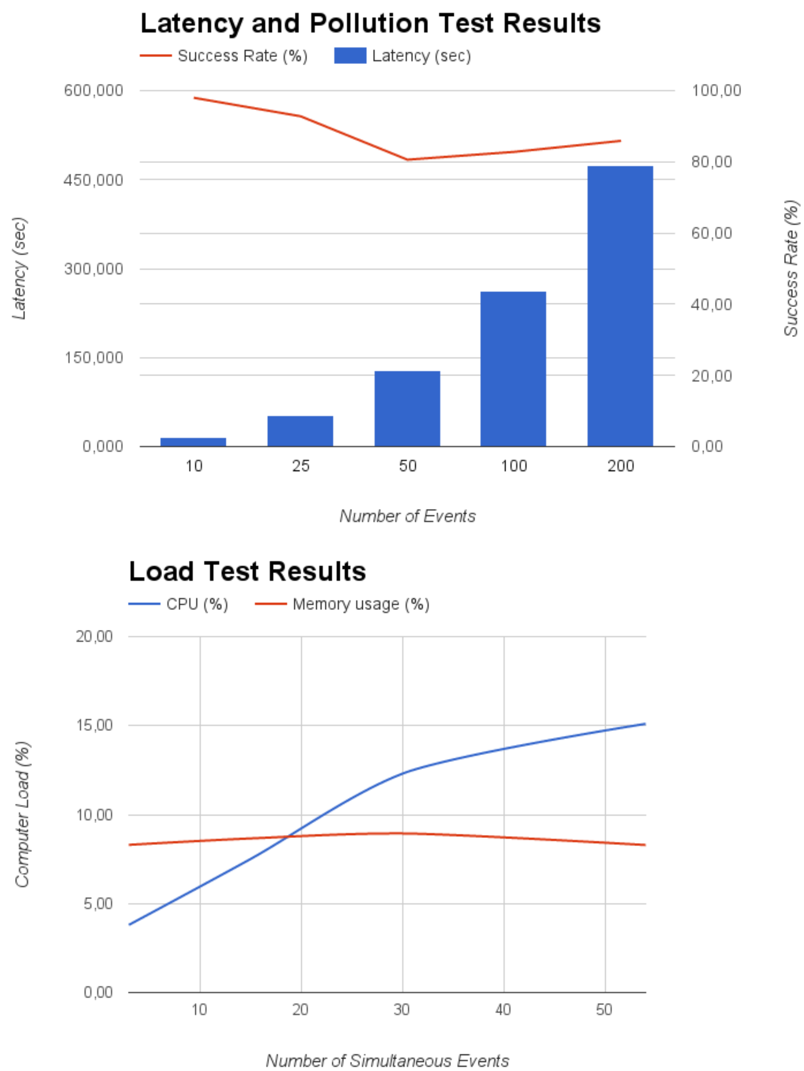

Figure 5.4. Latency, pollution and load test results.

In the Figure 5.4 two graphics can be observed. The first one (Latency and Pollution Test Results) illustrates that the latency represented by blue bars, behaves in a linear fashion regarding the number of events, which is to be expected since a larger quantity of events requires more time to be processed and analyzed by the prototype, on the other hand the success rate represented by a red line shows an initial decrease and a later stabilization, where the initial decrease was not due to the triggering of false alarms in the reports but the absence of expected reports, presumably caused by hardware limitations within the server, i.e. the server could not handle the increasing stream of incoming events generating the absence of some expected reports; notwithstanding, considering the application scenario the prototype should not handle more than 54 events every 15 minutes.

The second illustration (Load Test Results) shows two different behaviors, where the CPU Load represented by a blue line had a linear increase as expected since with a larger quantity of events the server must implement more physical resources to process and analyze them, but the Memory Usage represented by a red line is always around 9 % showing that Mobicents has an standardized memory consumption, which supports the previous idea where the server could not handle the increasing amount of events generating the decrease in the obtained success rate.

Considering the application scenario, i.e. the detection and warning of the coffee rust disease appearance probability in Colombian plantations, the obtained results evidence that the developed prototype is capable of fulfilling the performance requirements stated for the implementation of rural early warning systems in a developing country such as Colombia since the amount of climate data to be analyzed will not reach extreme levels that could cause a malfunctioning of the prototype.

Summary

This chapter presented a detailed description of the experimental prototype implemented in this undergraduate thesis, illustrating the deployment model on figure 5.1 and explaining its different components divided in the five modules previously presented in the architecture from section 1.5. Furthermore a set of tests were conducted on the experimental prototype specifying the test objectives, the implemented methodology and finally explaining the obtained results, which were illustrated in the figures 5.2, 5.3 and 5.4.

Chapter 6

Conclusions and future works

This chapter presents the conclusions obtained from the development of this undergraduate thesis along with some possible future works, proposed to improve the performance of the implemented platform.

6.1. Conclusions

In this section the main conclusions obtained from the development of this undergraduate thesis are presented as follows:

- After an extensive investigation and analysis of the related works of this undergraduate thesis, it can be concluded that although there are multiple approaches of early warning systems which implement converged environments in different scopes, there are not early warning systems that integrate complex event processing functionalities with converged environments to perform a monitoring of specific emergency situations.

- A system capable of performing the early detection and later notification of the coffee rust disease appearance probability integrated by complex event processing and converged environments was successfully developed aiming the improvement of the coffee farmers quality of life.

- A characterization and selection process of different event processing tools and converged environment platforms based on specific criteria was performed, concluding that the most suitable options to this undergraduate thesis conditions and objectives were Mobicents JAIN SLEE, Esper engine and OpenBTS tools.

- A set of event patterns was defined, taking as case study the prevention an early detection of the coffee rust disease appearance probability. The event patterns were classified into two categories: risky situations, separated in 3 different alert levels depending on the appearance probability value (low risk, medium risk, high risk). The second category is abnormal situations which represent a set of multiple malfunctioning events that the system can present such as null values, damaged hardware, data irregularities, among others.

- The Mobicents JAIN SLEE server is a feasible alternative for the development, deployment and execution of converged services based on open source software.

- From the obtained results analysed in chapter 5, it is possible to conclude that a successful integration between the converged environments supported by the JAIN SLEE specification and the Complex Event Processing supported by the Esper engine was achieved with the aim of identifying and notifying the coffee rust disease appearance probability through the real time analysis of crop related climate data, obtaining satisfactory results in both performed evaluations.

- From the obtained results analysed in chapter 5, it is possible to conclude that the most reliable services are the Email and the Twitter service. The fastest services are Email, Twitter and SMS, however the SMS service must be enhanced by improving the interaction between the converged service platform and the OpenBTS tools.

- The integration of the JAIN SLEE specification, OpenBTS tools and Complex Event Processing is a feasible alternative for the implementation of converged services in rural early warning systems, although the performance of the system can be improved by enhancing the hardware and software tools integrated within the developed platform, since most of the presented inconveniences were caused by the inability to handle large amount of events at the same time, generating the absence of some expected reports during the testing of the experimental prototype.

6.2. Future works

Considering the investigation area of this undergraduate thesis, the following future works are proposed:

- Update all the hardware tools integrated in the experimental prototype in order to obtain better performance results.

- Integrate some automatic mitigation processes once a low risk situation is detected, in order to release responsibilities from the users.

- Develop a web interface which enables the users registration and the selection of which alert types the user expects to receive.

- Develop an interface to facilitate the management of configuration parameters and all the subscribers information.

- Improve the interaction between the converged service platform and the telecommunications service dissemination tool, in order to enhance the success rate of the Telco services.

- Implement the experimental prototype in a real stage, integrating it with a meteorological station like the one used in [59] and the prediction module, in order to obtain information about its usability and performance.

- Realize a performance comparison between our proposal and a similar project implemented with commercial software and hardware tools in order to validate the viability for an actual implementation of a rural early warning system.

- Increase the number of dissemination and communication services within the converged control module in order to obtain a more effective notification process, with Web 2.0 and Telco services like Facebook, Google Translator, an AM or FM radio broadcast, among others.

- Integrate a response capability component within the proposed architecture in order to offer comprehension and preparation methods for the vulnerable population, providing emergency preparedness.

References

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

[2] Julián D. Vazquez, Iván F. Santa, y José V. Restrepo, «Prototipo de una Estación Celular Portatíl para Atención de Emergencias», p. 16, 2012.

[3] A. Hinze, K. Sachs, y A. Buchmann, «Event-based Applications and Enabling Technologies», en Proceedings of the Third ACM International Conference on Distributed Event-Based Systems, New York, NY, USA, 2009, pp. 1:1–1:15.

[4] Goro Kobayashi, Kazuhiko Fujita, «Convergence Service Platform», vol. 47, n.º 4, p. 8, 2011.

[5] «Complex Event Processing with Aspect Oriented Programming», p. 2, 2009.

[6] G. E. Churcher y J. Foley, «Applying Complex Event Processing and Extending Sensor Web Enablement to a Health Care Sensor Network Architecture», en Sensor Systems and Software, S. Hailes, S. Sicari, y G. Roussos, Eds. Springer Berlin Heidelberg, 2010, pp. 1-10.

[7] Julián Eduardo Plazas, Juan Sebastián Rojas, Juan Carlos Corrales, «An Architecture Based on JAIN SLEE and OpenBTS for Rural Early Warning Systems», presentado en CITA 2015, Popayán, Colombia, 2015, p. 6.

[8] «Congreso Iberoamericano de Telemática», http://cita2015.co/, 2015. [En línea]. Disponible en: http://cita2015.co/.

[9] Julián Eduardo Plazas, Juan Sebastián Rojas, Juan Carlos Corrales, «Improving Rural Early Warning Systems Through The Integration of OpenBTS and JAIN

SLEE», 2016, n.º 29, p. 20, 2016.

[10] «Revista Ingenierías Universidad De Medellín», http://www.scielo.org.co/scielo.php?script=sci_serial&pid=1692-3324&lng=en&nrm=iso. [En línea]. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_serial&pid=1692-3324&lng=en&nrm=iso.

[11] J. Rojas, L. Ordóñez-Ante, y J. Carlos, «Web and Telco Service Integration: A Dynamic and Adaptable Approach», Int. J. Adv. Comput. Sci. Appl., vol. 5, n.º 5, 2014.

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

[13] «JAIN - Reference.» [En línea]. Disponible en: http://www.oracle.com/technetwork/java/index-137964.html.

[14] «JSR-000240 JAIN SLEE 1.1 - Final Release.» [En línea]. Disponible en: https://jcp.org/aboutJava/communityprocess/final/jsr240/.

[15] J. R. E. Schooler, «SIP: Session Initiation Protocol», RFC3261. [En línea]. Disponible en: https://www.ietf.org/rfc/rfc3261.txt.

[16] «OpenBTS», Wikipedia, the free encyclopedia. 03-feb-2015.

[17] «OpenBTS Public Release.» [En línea]. Disponible en: http://openbts.org/w/index.php/Main_Page.

[18] «Developing Early Warning Systems: A Checklist», presentado en EWC III Third International Conference on Early Warning, Bonn, Germany, 2006, p. 10.

[19] B. Van Den Bossche, S. Van Hoecke, C. Danneels, J. Decruyenaere, B. Dhoedt, y F. De Turck, «Design of a JAIN SLEE/ESB-based platform for routing medical data in the ICU», Comput. Methods Programs Biomed., vol. 91, n.º 3, pp. 265-277, sep. 2008.

[20] J. Li, Q. Li, S. U. Khan, y N. Ghani, «Community-based cloud for emergency management», en 2011 6th International Conference on System of Systems Engineering (SoSE), 2011, pp. 55-60.

[21] A. Ordonez, V. Alcázar, J. C. Corrales, y P. Falcarin, «Automated context aware composition of Advanced Telecom Services for environmental early warnings», Expert Syst. Appl., vol. 41, n.º 13, pp. 5907-5916, oct. 2014.

[22] Taha Hajar Tarek Sakakini y Wajeb Saab, «NI USRP GSM Base Station.»

[23] K. Heimerl, K. Ali, J. Blumenstock, B. Gawalt, y E. Brewer, «Expanding Rural Cellular Networks with Virtual Coverage», en Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, Berkeley, CA, USA, 2013, pp. 283–296.

[24] J. Mpala y G. van Stam, «Open BTS, a GSM Experiment in Rural Zambia», en e-Infrastructure and e-Services for Developing Countries, K. Jonas, I. A. Rai, y M. Tchuente, Eds. Springer Berlin Heidelberg, 2013, pp. 65-73.

[25] R. A. Ali, A. N. Khan, S. Arshad, y U. Younis, «Towards the development of LTE networks: Implementation of OpenIMSCore in asterisk/OpenBTS GSM network», en 2013 International Conference on Open Source Systems and Technologies (ICOSST), 2013, pp. 103-106.

[26] «Femtocelda», Wikipedia, La enciclopedia libre. 27-jun-2015.

[27] «Estación base», Wikipedia, la enciclopedia libre. 11-nov-2014.

[28] «Software libre», Wikipedia, la enciclopedia libre. 15-jun-2015.

[29] Nuno Silva, Marco Monteiro, Sancho Rego, Herlander Jorge, «Service Oriented Architectures for Convergent Service Delivery Platforms.» Pierre-Arnaud Muller, France Télécom Group, 2006.

[30] «Oracle Communications Converged Application Server.» 2010.

[31] «IBM - Service Provider Delivery Environment Framework», 14-mar-2012. [En línea]. Disponible en: http://www-01.ibm.com/software/industry/communications/framework/. [Accedido: 14-jul-2015].

[32] «Ericsson Service Delivery Platform.» .

[33] «Mobicents Media Server.» [En línea]. Disponible en: http://www.mobicents.org/mms/mms-main.html. [Accedido: 14-jul-2015].

[34] «Rhino Documentation - Dev Portal - OpenCloud Developer’s Portal.» [En línea]. Disponible en: https://developer.opencloud.com/devportal/display/OCDEV/Rhino+Documentation. [Accedido: 14-jul-2015].

[35] «Parlay», Wikipedia, la enciclopedia libre. 21-jun-2014.

[36] «Subsistema Multimedia IP», Wikipedia, la enciclopedia libre. 09-nov-2014.

[37] «Operations Support Systems», Wikipedia, la enciclopedia libre. 23-abr-2014.

[38] «Session Initiation Protocol», Wikipedia, la enciclopedia libre. 27-abr-2015.

[39] «Hypertext Transfer Protocol», Wikipedia, la enciclopedia libre. 29-jun-2015.

[40] «Diameter (protocolo)», Wikipedia, la enciclopedia libre. 18-ene-2015.

[41] «EsperTech - Event Series Intelligence.» [En línea]. Disponible en: http://www.espertech.com/. [Accedido: 14-jul-2015].

[42] «Getting started with Esper in 5 minutes.», CoffeeOneSugar's Blog. .

[43] «Microsoft StreamInsight.» [En línea]. Disponible en: https://technet.microsoft.com/en-us/library/ee362541(v=sql.111).aspx. [Accedido: 14-jul-2015].

[44] «Microsoft® SQL Server® StreamInsight 2.0», Microsoft Download Center. [En línea]. Disponible en: https://www.microsoft.com/en-us/download/details.aspx?id=29070. [Accedido: 14-jul-2015].

[45] «SQL Server 2014 | Microsoft.» [En línea]. Disponible en: https://www.microsoft.com/en-us/server-cloud/products/sql-server/purchasing.aspx. [Accedido: 14-jul-2015].

[46] «Apama Streaming Analytics | Software AG.» [En línea]. Disponible en: http://www.softwareag.com/corporate/products/apama_webmethods/analytics/overview/default.asp. [Accedido: 14-jul-2015].

[47] «Sell-Side Technology Awards 2014.» 2014.

[48] «Altibase», Wikipedia, the free encyclopedia. 17-abr-2015.

[49] «What is an In-Memory Database? | Altibase.» [En línea]. Disponible en: http://altibase.com/what-is-in-memory/. [Accedido: 14-jul-2015].

[50] «ALTIBASE Online Documentation.» [En línea]. Disponible en: http://support.altibase.com/manual/en/533b/html/altiman.html. [Accedido: 14-jul-2015].

[51] «Telecommunications Archives», Altibase. [En línea]. Disponible en: http://altibase.com/case-category/telecommunications/. [Accedido: 14-jul-2015].

[52] «Oracle Complex Event Processing.» [En línea]. Disponible en: http://www.oracle.com/technetwork/middleware/complex-event-processing/overview/complex-event-processing-088095.html. [Accedido: 14-jul-2015].

[53] «Oracle JDeveloper - Official Home Page.» [En línea]. Disponible en: http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.htm. [Accedido: 14-jul-2015].

[54] «Drools - Drools - Business Rules Management System (Java™, Open Source).» [En línea]. Disponible en: http://www.drools.org/. [Accedido: 24-jul-2015].

[55] «Drools Documentation.» [En línea]. Disponible en: http://docs.jboss.org/drools/release/6.2.0.Final/drools-docs/html/index.html. [Accedido: 24-jul-2015].

[56] «Drools Fusion and Utilization of Complex Event Processing in Web Applications.» [En línea]. Disponible en: https://theses.cz/id/1qszog?lang=en;furl=%2Fid%2F1qszog;so=nx. [Accedido: 24-jul-2015].

[57] Alexander Wahl, Bernhard Hollunder, «Performance Measurement For CEP Systems», presentado en Service Computation 2012, The Fourth International Conferences on Advanced Service Computing, Nice, France, 2012, p. 6.

[58] D. C. Corrales, A. J. P. Q, C. León, A. F. Casas, y J. C. Corrales, «Early warning system for coffee rust disease based on error correcting output codes: a proposal», Rev. Ing. Univ. Medellín, vol. 13, n.º 25, p. 6-, 2014.

[59] D. C. Corrales, A. Ledezma, A. J. P. Q, J. Hoyos, A. Figueroa, y J. C. Corrales, «A new dataset for coffee rust detection in Colombian crops base on classifiers», Sist. Telemática, vol. 12, n.º 29, pp. 9-23, jun. 2014.

[60] O. Luaces, L. H. A. Rodrigues, C. A. Alves Meira, y A. Bahamonde, «Using nondeterministic learners to alert on coffee rust disease», Expert Syst. Appl., vol. 38, n.º 11, pp. 14276-14283, oct. 2011.

[61] M. E. Cintra, C. A. A. Meira, M. C. Monard, H. A. Camargo, y L. H. A. Rodrigues, «The use of fuzzy decision trees for coffee rust warning in Brazilian crops», en 2011 11th International Conference on Intelligent Systems Design and Applications (ISDA), 2011, pp. 1347-1352.

[62] «emergency - definition of emergency in English from the Oxford dictionary.» [En línea]. Disponible en: http://www.oxforddictionaries.com/definition/english/emergency. [Accedido: 14-

jul-2015].

[63] «Types of Disaster - Natural and Man-Made - RestoreYour Economy», Restore Your Economy. .

[64] A. C. Kushalappa y A. B. Eskes, Coffee Rust: Epidemiology, Resistance, and Management. CRC Press, 1989.

[65] La Roya del Cafeto en Colombia Impacto, manejo y costos del control, 2011.[a] ed. .

[66] «El paloteo o die-back (muerte regresiva) en café.» [En línea]. Disponible en: http://sian.inia.gob.ve/repositorio/revistas_tec/FonaiapDivulga/fd50/paloteo.htm. [Accedido: 14-jul-2015].

[67] C. A. A. Meira, L. H. A. Rodrigues, y S. A. Moraes, «Analysis of coffee leaf rust epidemics with decision tree», Trop. Plant Pathol., vol. 33, n.[o] 2, pp. 114-124, abr. 2008.

[68] «EsperTech - Products - Esper.» [En línea]. Disponible en: http://www.espertech.com/products/esper.php. [Accedido: 14-jul-2015].

[69] «Service provider», Wikipedia, the free encyclopedia. 06-abr-2015.

[70] «Service catalog», Wikipedia, the free encyclopedia. 07-jul-2015.

[71] «JAIN SLEE User Guide.» [En línea]. Disponible en: http://docs.jboss.org/mobicents/jain-slee/2.7.0.FINAL/container/user-guide/en-US/html/. [Accedido: 19-jun-2015].

[72] D. C. Corrales, A. Figueroa, A. Ledezma, y J. C. Corrales, «An Empirical Multi-classifier for Coffee Rust Detection in Colombian Crops», en Computational Science and Its Applications -- ICCSA 2015, vol. 9155, O. Gervasi, B. Murgante, S. Misra, M. L. Gavrilova, A. M. A. C. Rocha, C. Torre, D. Taniar, y B. O. Apduhan, Eds. Cham: Springer International Publishing, 2015, pp. 60-74.

[73] «GNU Radio Manual and C++ API Reference: Build Instructions and Information.» [En línea]. Disponible en: http://gnuradio.org/doc/doxygen/build_guide.html. [Accedido: 15-jul-2015].

[74] «Index of /releases/gnuradio.» [En línea]. Disponible en: http://gnuradio.org/releases/gnuradio/. [Accedido: 15-jul-2015].

[75] «Setup USRP N210 - Byungchul's Archive.» [En línea]. Disponible en: https://sites.google.com/site/byungchulweb/setup-usrp2. [Accedido: 15-jul-2015].

[76] «RadioIntegration - OpenBTS.» [En línea]. Disponible en: http://openbts.org/w/index.php/RadioIntegration#Universal_Hardware_Driver. [Accedido: 15-jul-2015].

[77] «BuildInstallRun - OpenBTS.» [En línea]. Disponible en: http://openbts.org/w/index.php/BuildInstallRun. [Accedido: 15-jul-2015].

[78] «MultiBTS - OpenBTS.» [En línea]. Disponible en: http://openbts.org/w/index.php/MultiBTS. [Accedido: 15-jul-2015].

[79] «Generating SSH keys - User Documentation.» [En línea]. Disponible en: https://help.github.com/articles/generating-ssh-keys/. [Accedido: 15-jul-2015].

[80] «How to configure a HTTP proxy for svn - Stack Overflow.» [En línea]. Disponible en: http://stackoverflow.com/questions/1491180/how-to-configure-a-http-proxy-for-svn. [Accedido: 15-jul-2015].

Appendix A

OpenBTS installation guide

This appendix presents an OpenBTS installation guide with a USRP N-210, completed by recollecting and organizing information from various OpenBTS-related web sites containing important but scattered information. It is worth mentioning that the steps described here where implemented in a 32 bits Ubuntu 12.04 following the OpenBTS recommendations and it should be noted that to have a successful connection, the computer must have a network card of a 1GB minimum speed.

All the commands will be written between brackets which are not included in the command itself and all the needed explanations are described in each step. The process begins with the installation of GNURadio and UHD, then it proceeds with the installation of OpenBTS software.

A.1. GNURadio and UHD installation process

Install Synaptic Package Manager from the Ubuntu Software Center.

1. GNURadio needs some dependencies that must be install before GNURadio itself. The dependencies are: git, a cmake version above 2.6.3, a boost version

above 1.35, a cppunit version above 1.9.14 and a fftw3f version above 3.0.1 [73].

To install boost, search in the Ubuntu software center the library "*libboost-all-dev*" implement the A.1 command:

(A.1) {sudo apt-get install python libboost-all-dev libusb-1.0-0-dev}.

In order to observe the installed version, enter the command A.2:

(A.2) {dpkg -s $PackageName | grep 'Version'}.

2. Deactivate the auto-lock option in the computer, since the installation process may be interrupted.

3. Perform and update and upgrade to the Ubuntu 12.04 libraries with the A.3 commands:

(A.3a) {sudo apt-get update}.
(A.3b) {sudo apt-get upgrade}.

4. Install UHD through Synaptic Package Manager.

5. Perform and update and upgrade to the Ubuntu 12.04 libraries with the A.3 commands from step 3.

6. Download GNURadio 3.6.0 comprised in a tar.gz file from its web page (http://gnuradio.org/releases/gnuradio/). Later versions have a conflict finding libboost libraries [74].

7. Create a new folder to store the GNURadio source code.

8. Move the GNURadio comprised file to the folder created in step 7.

9. Access to the folder through the command line and enter the A.4 commands:

(A.4a) {mkdir build}
(A.4b) {cd build}
(A.4c) {cmake ../}
(A.4d) {make}
(A.4e) {make test}
(A.4f) {sudo make install}

This process may take a while, however if it takes too much time it is better to cancel the process and restart it.

10. Specify the environment variables “*pythonpath*” and “*ld_library_path*” pasting the following lines in the Ubuntu *bashrc* file:

export PYTHONPATH=$PYTHONPATH:/usr/lib/python2.7
export LD_LIBRARY_PATH=$LD_LIBRARY__PATH:/usr/local/lib

Verify that the files are precisely in the given folder directions.

11. Finally it is possible to check if the installation was successfully completed by entering the A.5 command:

(A.5) {gnuradio-companion}

Which displays the GNURadio environment. Also the UHD option should appear among the environment’s resources.

12. To test that the computer can connect to the USRP-N210, first it is necessary to connect the computer to the USRP through the Ethernet cable and connect the antenna to the USRP. Then a new wired network configuration profile must be created assigning the 192.168.10.3 IP address to the computer with a 255.255.255.0 subnet mask and without a gateway, since that is how the USRP-N210 is configured [75] [76].

Also it is necessary to remove the proxy (if the computer has any). Remember that to have a successful connection, the computer must have a network card of a 1GB minimum speed. Finally on the command line it is possible to perform two tests.

The first one is sending a ping to the 192.168.10.2 address which is the default address assigned to the USRP. Once this test is successfuly completed, enter the A.6 command:

(A.6) {uhd_usrp_probe}

This command broadcasts a probe in order to verify if there are any UHD equipment available. Once this command is executed the obtained result should display a set of messages reporting the USRP N-210 availability.

A.2. OpenBTS software installation process

This section describes the installation process of the OpenBTS software with several other tools needed for its performance (Asterisk, Transceiver, SipAuthServe, Smqueue and RripServer).

Also this installation process only describes the configuration of a unique instance of OpenBTS with a singular computer and USRP. This guide does not handle the multiple BTS installation process [77] [78].

1. Install Asterisk PBX from Synaptic Package Manager.

2. Create a folder to store the subscriber registry entering the A.7 command:

   (A.7) {sudo mkdir -p /var/lib/asterisk/sqlite3dir}.

3. Create a folder to store the OpenBTS source code.

4. Access such folder through the command line, also it is necessary to update GIT to its latest version entering the A.8 commands:

(A.8a) {sudo apt-get install software-properties-common python-software-properties}
(A.8b) {sudo add-apt-repository ppa:git-core/ppa}
(A.8c) {sudo apt-get update}
(A.8d) {sudo apt-get install git}

Verify that the GIT version is above the 1.8.2, for this enter the A.9 command:

(A.9) {git --version}

5. Check if there are SSH keys using the A.10 command:

(A.10) {ls -al ~/.ssh}

If there are not SSH keys proceed to step 6, otherwise proceed with the step.

6. If there are not SSH keys then create one with your email using the A.11 command:

(A.11) {ssh-keygen -t rsa -C "tu_email@example.com"}

It is recommended to use the same email assigned to your GitHub account. The creation of the key asks to enter a file, leave this field blank. A password key (passphrase) is requested, it is recommended to create a safe one [79].

7. To add the created key to the SSH controller use the A.12 commands:

(A.12a) {eval "$(ssh-agent -s)"}
(A.12b) {ssh-add ~/.ssh/id_rsa}

8. Once you have your own ssh key, it must be added to Github. To accomplish this follow the next steps:

Install xclip from the repository with the A.13 command:

(A.13) {sudo apt-get install xclip}.

Copy your SSH key with xclip using the A.14 command:

(A.14) {xclip -sel clip < ~/.ssh/id_rsa.pub}.

Access the Github web page and log in, if you do not have an account, create one with the mail that provided in the step 5.

Click Settings in the upper right corner. Select the SSH keys option and click "*Add SSH key*".

Enter a title to identify the computer, e.g. OpenBTS-Server.

Paste your previously created SSH key and finally click on "*Add key*".

9. Verify all the previous steps were successfuly completed through the A.15 command:

(A.15) {ssh -T git@github.com},

This will display a question that must be answered affirmatively.

10. Download the OpenBTS source code using the A.16 commands:

(A.16a) {git clone https://github.com/RangeNetworks/dev.git}
(A.16b) {cd dev}
(A.16c) {./clone.sh}

11. Use the A.17 command to select the needed versión, writing "master" to select the latest version or "v4.0.0" for the latest stable version (September 2014).

(A.17a) {./switchto.sh $version}

For example:

(A.17b) {./switchto.sh v4.0.0}

12. Use the A.18 command to install the libraries needed to run OpenBTS

(A.18) {sudo apt-get install autoconf libtool libosip2-dev libortp-dev libusb-1.0-0-dev g++ sqlite3 libsqlite3-dev erlang libreadline6-dev libncurses5-dev}

13. Install SubVersion using the A.19 command:

(A.19) {sudo apt-get install subversión}

In order to use SubVersion within a proxy it is necessary to access the “servers” file from the SubVersion folder through the A.20 command:

(A.20) {sudo gedit .subversion/servers}

In this file go to the “global” context and add the following lines, in this example the proxy belongs to the University of Cauca [80]:

http-proxy-host = proxy.unicauca.edu.co
http-proxy-port = 3128
http-proxy-compression = no

It is worth mentioning that this process is not needed if the network does not have a proxy.

14. Run the Update and Upgrade commands again.

15. Another library that OpenBTS needs is liba53 included in the distribution; use the A.21 commands to install it:

(A.21a) {cd liba53}
(A.21b) {sudo make install}
(A.21c) {cd ../}

The last command returns to the OpenBTS main folder.

16. To install the OpenBTS software the “*Build.sh*” script must be executed using the A.22 command:

(A.22) {./build.sh N210}

Running this command configures some components are installed to be taking into consideration the implemented hardware (USRP N210).

17. To configure the Transceiver enter the A.23 commands:

(A.23a) {cd openbts}
(A.23b) {autoreconf -i}
(A.23c) {./configure --with-uhd}
(A.23d) {make}

18. To create a symbolic link with the Transceiver enter the A.24 commands:

(A.24a) {cd apps}
(A.24b) {ln -s ../Transceiver52M/transceiver .}
(A.24c) {cd ../}

19. Configure OpenBTS database using the A.25 commands:

(A.25a) {sudo mkdir /etc/OpenBTS}
(A.25b) {sudo sqlite3 -init ./apps/OpenBTS.example.sql /etc/OpenBTS/OpenBTS.db ".quit"}

20. Verify that step 18 was successfully completed executing the A.26 command; it should display a configuration of SQL variables.

(A.26) {sqlite3 /etc/OpenBTS/OpenBTS.db .dump}

21. Reconnect the USRP to the computer using the settings made in the step 13 of the GNURadio and UHD installation process.

22. In order to verify that everything has been performed correctly, start OpenBTS with the USRP up and running, following the A.27 commands:

(A.27a) {cd apps}
(A.27b) {sudo ./OpenBTS}

The previous commands start OpenBTS. If everything is properly configured the command line should display a “system ready” message. This message is usually displayed at the end of the command line. If it is not displayed then the USRP is not properly connected or the installation process failed in some of the previous steps.

23. Proceed with the installation and configuration process of SipAuthserve the user registration tool in the OpenBTS architecture. First, create the folder which will store the subscribers registration using the A.28 command:

(A.28) {sudo mkdir -p /var/lib/asterisk/sqlite3dir}

Now Access the “subscriberRegistry” folder stored in the OpenBTS folder through the A.29 command:

(A.29) {cd suscriberRegistry}

Inside this folder, proceed with the A.30 installation and configuration commands:

(A.30a) {autoreconf -i}
(A.30b) {./configure}
(A.30c) {make}

Once this process finishes, an executable file should be inside the “*subscriberRegistry/apps*” folder.

24. Now proceed to create the subscribers database inside the "*apps*" folder following the A.31 command:

(A.31) {sudo sqlite3 -init ./apps/subscriberRegistry.example.sql /etc/OpenBTS/sipauthserve.db ".quit"}

To execute SipAuthserve enter the A.32 commands:

(A.32a) {cd apps}
(A.32b) {sudo ./sipauthserve}

It is worth mentioning that SipAuthserve command line must not be modified after the execution.

25. Now proceed with the messaging server (Smqueue) installation process. Enter into the server's folder that is inside the OpenBTS directory with the A.33 command, as follows:

(A.33) {cd smqueue}

Inside this folder, we must perform a similar installation process to the SipAuthserve, illustrated through the A.34 commands:

(A.34a) {autoreconf -i}
(A.34b) {./configure}
(A.34c) {make}

26. Now proceed to create the database that will store the messaging server configuration with the A.35 command:

(A.35) {sudo sqlite3 -init smqueue/smqueue.example.sql /etc/OpenBTS/smqueue.db ".quit"}

27. Before executing the messaging server, a directory with the following address "*/var/lib/OpenBTS*" must be created, the creation of these folders is necessary

to create a file which will store the record of the sent messages with the A.36 command:

(A.36) {sudo mkdir /var/lib/OpenBTS}

28. To run the Smqueue messaging server using A.37 commands, the execution command must be entered in the folder of the same name, it should be noted that Asterisk PBX must be configured first.

(A.37a) {cd smqueue}
(A.37b) {sudo ./smqueue}

The previous command displays an “ALERT” message to illustrate the starting process of the server.

29. To activate the GPRS service the next simple steps must be followed:

Access the OpenBTS Command Line through the A.38 commands:

(A.38a) {cd openbts/apps}
(A.38b) {sudo ./OpenBTSCLI}

Activate the GPRS service within the OpenBTS Command Line through the A.39 commands:

(A.39) {config GPRS.Enable 1}

Once this process is finished the OpenBTS CLI can be closed.

Observe the computer’s network interfaces entering the A.40 command in other command line:

(A.40) {ifconfig}.

Identify which interface access the internet and which interface is used for the USRP connection. Usually the internet interface is named “wlan0” or “eth0”. For this guide the internet interface is identified by “eth0”.

Implement the preferred text editor (Gedit in this case) to access the “*iptables.rules*” file within the “*apps*” folder, stored inside the OpenBTS directory. To start Gedit enter the A.41 command:

(A.41) {gedit iptables.rules}

Within the “*iptables.rules*” file, find the following line:

-A POSTROUTING -o wlan0 -j MASQUERADE

Change "wlan0" for the internet interface identified previously (eth0) and save the file. The found interface (wlan0) can change depending on the computer’s configuration.

In this case the “*iptables.rules*” file was changed as illustrated subsequently:

-A POSTROUTING -o eth0 -j MASQUERADE

With this we can close Gedit.

Save the routing table in the OpenBTS configuration, entering the A.42 command in the “*openbts/apps*” directory:

(A.42) {sudo iptables-restore < iptables.rules}

Save the OpenBTS network configuration in the computer so it can be used every time the computer is restarted, this can be done through the A.43 commands:

(A.43a) {cd /etc/network}

(A.43b) {sudo pre-up iptables-restore <
$OpenBTS_source_folder/dev/openbts/apps/iptables.rules}

Now a DNS server must be established, this process is quite simple. Access through Gedit the “*resolv.conf*” file stored inside the “*etc*” folder, like the A.44 command:

(A.44) {sudo gedit /etc/resolv.conf}

Change the "*nameserver*" attribute to 8.8.8.8 and save the file. The attribute must be like this:

nameserver 8.8.8.8

Now save this configuration so it keeps working after every computer’s restart. For this edit the “*head*” file stored in “*etc/resolvconf/resolvconf.d/*” directory. Enter the A.45 command to perform this change:

(A.45) {sudo gedit /etc/resolvconf/resolvconf.d/head}

30. Finally, it is necessary to shut down the computer’s firewall in order to offer internet access to the GSM phones. For this enter the A.46:

(A.46) {sudo ufw disable}

To restart the firewall enter the A.47 command:

(A.47) {sudo ufw enable}.

A.3 Testing OpenBTS, Asterisk and the USRP N210

1. To perform voice calls and send SMS, first start all the tools previously installed (OpenBTS, OpenBTSCLI, Asterisk, SipAuthserve and Smqueue), in different command line tabs, entering the start commands described in the installation process.

It is worth mentioning that before starting all the tools, the USRP N210 must be already connected to the computer.

2. Now we must enable the connection of GSM devices to the network, since the network is unable to connect with the phones by default. Access the OpenBTS Command Line and proceed with the configuration of the Open Registration variable as A.48 command:

   (A.48) {config Control.LUR.OpenRegistration .*}

3. Once all the needed tools are up and running, proceed to connect a GSM phone to the network. Usually any phone with a GSM SIMCARD should have the "*network operators*" option in the configuration panel. Start a network searching and access the network called "*00101*" or "*Range*" (sometimes the name changes), this will connect the GSM phone to the OpenBTS network

4. After some time the GSM phone is connected to the network will receive a SMS from the 101 with its IMSI (International Mobile Subscriber Identity) which will be needed in the following steps for the voice call configuration.

5. Once the message is received, answer with the phone number that will be assigned to the GSM phone for the SMS configuration.

6. Send a SMS with any text to the 411 after the phone number has been registered. Then the phone will receive a response with its IMSI and assigned phone number.

7. Once the 411 response is received, the Asterisk files "*Sip.conf*" and "*Extensions.conf*" must be configured in order to perform voice calls implementing the GSM protocol. For this the lines on Figures A.1 and A.2 must be added in the mentioned files (it is worth mentioning that the observed IMSI, phone number and callerid must be replaced for the ones received in the 101 response).

```
[IMSI732XXXXXXXXXX57]
callerid=3115642574
canreinvite=no
type=friend
allow=gsm
context=sip-external
host=dynamic
dtmfmode=info

[IMSI732XXXXXXXXXX53]
callerid=3015232524
canreinvite=no
type=friend
allow=gsm
context=sip-external
host=dynamic
dtmfmode=info
```

Figure A.1. Sip.conf configuration.

```
[macro-dialGSM]
exten => s,1,Dial(SIP/${ARG1},20)
exten => s,2,Goto(s-${DIALSTATUS},1)
exten => s-CANCEL,1,Hangup
exten => s-NOANSWER,1,Hangup
exten => s-BUSY,1,Busy(30)
exten => s-CONGESTION,1,Congestion(30)
exten => s-CHANUNAVAIL,1,playback(ss-noservice)
exten => s-CANCEL,1,Hangup
[sip-external]
exten =>
3115642574,1,Macro(dialGSM,IMSI732XXXXXXXXXX57@127.0.0.1:5062)
exten =>
3015232524,1,Macro(dialGSM,IMSI732XXXXXXXXXX53@127.0.0.1:5062)
```

Figure A.2. Extensions.conf configuration.