Design and Implementation of a Tool to Collect Data of a Smart City Through the TV

Glaydstone Alves Teixeira
CESAR – Recife Center for Advanced Studies and Systems
Centro de Estudos e Sistemas Avançados do Recife
Recife, Brazil
glaydstone.a.teixeira@gmail.com

Felipe Silva Ferraz
CESAR – Recife Center for Advanced Studies and Systems
Centro de Estudos e Sistemas Avançados do Recife
Recife, Brazil
fsf@cesar.org.br

Abstract — **Smart City is the future of urban planning of the cities in the next generations. In Smart Cities, values are added to services, such as optimizing traffic and better use of energetic resources, generating a resource saving. These services are a result of a wide data collection from sensors or through crowdsensing. The objective of this work is to propose the development of a tool called PlugTV to collect information of citizens through the TV. The tool is composed by a Web component, implemented as a service, which allows data collection as a log from heterogeneous clients (TV, connected devices and sensors) and by a mobile component, implemented in an Android Mini-PC attached to the TV that allows data collection from the TV in a transparent way for the citizen, abstracting the dependency on a Digital TV middleware. In parallel, there is a gain of portability when it comes to IPTV providers because each Smart TV produces has its own platform of development and communication.**

Keywords- mini-pc; middleware; crowdsensing.

I. INTRODUCTION

Seventy percents of the world population, i.e., 7.2 billion people are expected to be living in cities and surrounding areas by the year 2050 [1][2][3]. Cities are having more control on their political, technological and economic development due to growing urbanization. Parallel to it, they face a series of challenges and threats to sustainability in all their central systems, which need to be approached in a holistic way [4].

Washburn and Sindhu [5] affirm that cities are becoming “smarter” once governments, companies and communities rely more and more on technology to overcome the challenges of fast urbanization. What makes a “smart city” is the combined use of systems, software infrastructure and a network of interconnected devices – the so called smart computing technology – in order to connect better seven components and infrastructure services of the city: administration, education, health, public safety, real estate, transportation and public services.

A Smart City is instrumented and tracked by a group of sensors and devices that collect data in such a way they can dynamically measure the urban activities of the city, helping the infrastructure of them [6]. This is the role of the smart software in smart cities. In this scenery, the paradigm of the Internet of Things (IoT) is based on the identification and use of a great number of physical and virtual objects disposed in a heterogeneous way and connected to the Internet.

IoT is now a subject of intense research. Technologies of the IoT are being implemented in a great number of applications. In the scope of Smart City initiatives, systems based on IoT are playing an important role, allowing the use of network infrastructures to introduce or improve a variety of services for the citizens. A network sensor is a key element in the Internet of Things, therefore, a key element in the Smart City applications. A variety of data is collected through sensors distributed all over the city or through crowdsensing. Data collected include typical information about the traffic, energy consumption of houses and apartments, devices connected to the Internet, etc. These data are stored as logs and analyzed by mining techniques, transforming the information into services that may be useful for the lives of citizens. However, the cost to implement and maintain these sensors is considerably high. Dohler and Ratti affirm that citizens will be the living sensors of smart cities and the central systems of the city should be connected to them [7].

This article proposes the development of a tool called ***PlugTV*** to collect data of the urban environment of a city and information about the routine of the citizen through the TV and provide the information collected as raw data into more accessible data, delivering information of much aggregated value.

This paper is structured as follows: Section II discusses state of the art. Our tool, its uses and functionalities are described in section III, while section IV describes technologies applied. Our studies and future works are presented in section V and section VI provides concluding remarks.

II. BACKGROUND

A. Smart City

Smart Cities are those that use advanced technologies to find solutions for their problems and for the new demands of

 ISBN: 978-1-61208-438-1

the population. What makes a city smart is the combined use of software, network infrastructure and client devices [6]. In the smart city, a variety of data is collected from sensors and people all over the city. These data include information about the traffic, energy consumption in houses and apartments, use of home appliances, etc.

Data are stored as logs and analyzed by advanced techniques of data processing, then used to aggregate value to services for a sustainable society.

According to Dirks and Keeling, the city is a like a system of systems [8]. No system works in an isolated way; instead, there is an interconnecting net. For example, transportation, industry and energy systems are closely related – transportation and industry are the main users of energy. Connecting these systems will offer more efficiency for the sustainability in the long run. The connection between the water and energy systems is another example of the connections there are between systems. A substantial amount of electricity generated goes to the pumping and treatment of water. In Malta, for example, a new smart utility system will inform citizens and companies about the use of energy and water, allowing them to make better decisions on the consumption of resources.

B. IOT

Internet of Things (IoT) and Cloud Computing are nowadays two of the most popular paradigms in Information and Communication Technologies (ICT) and should build the next computing era [9]. The IoT allows the communication between different objects, as well as the context of service innovation towards applications with greater aggregated value.

According to JIN et al [10], the network environment in IoT is strongly characterized by heterogeneity. Heterogeneous networks have a multi-service platform, providing different possibilities of services and applications. Cities are composed by a set of complex and heterogeneous systems of different kinds: infrastructure of civil engineering, ICT infrastructure, social media, financial networks, etc. All systems demand great management effort (tracking, reports and interventions) to guarantee the constant performance of relevant activities and services [11].

In 2008, the number of “things” connected to the Internet exceeded the number of people on Earth and by 2020 this number will exceed 50 billion things connected to the Internet, as seen in the following figure, presented by CISCO [12]:

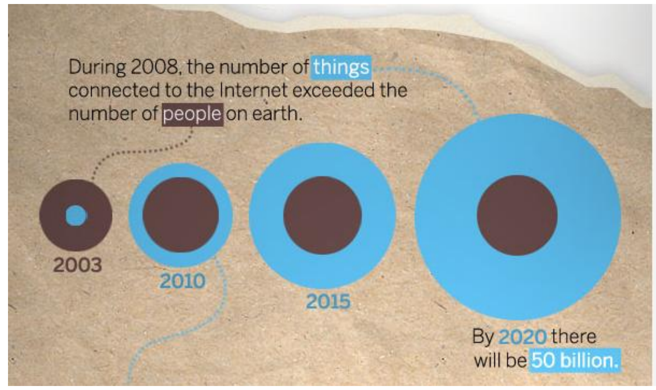

Figure 1. Growth of connected devices

As shown in Figure 2, the number of houses with TVs is three times higher in developing countries than in developed ones. When compared to the rest of the world, these numbers come to 1.3 higher. The discussed tool is extensible to treat heterogeneous data from different devices connected to the Internet, such as sensors, light, refrigerator, smart meters of water and energy, etc. The constant growth of the number of TVs in homes has motivated us to put the TV as the device used on this research for the proposed platform.

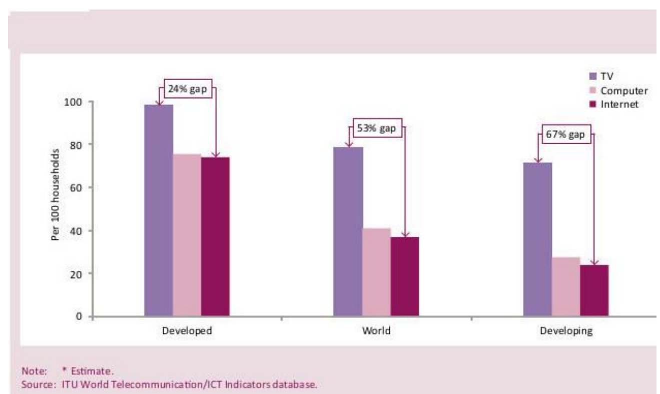

Figure 2. Growth of connected devices

C. Crowdsensing

Crowdsourcing is a new paradigm of managing information collection knowledge from a group of users, in order to execute complex tasks [13]. A well-known example of a crowdsourcing activity is Waze [14], one of the biggest traffic applications of the world based in a community.

III. PLUGTV

A. Architecture of the PlugTV

An application to collect data from the routine of the citizen through the TV is being developed. These data will be converted into useful service to the citizen. Figure 3 shows the general architecture of the PlugTV. Components of Figure 3 are the Smart City services, mobile component and WEB component, called PlugTVMobile and PlugTVRestful, respectively.

PlugTVMobile is implemented in an Android Mini-PC connected to the TV and it is responsible for collecting information from the routine of the citizen. When the TV is connected, the application creates an event via Broadcast notifying the change of status of the TV and this change is persisted through services disposed in the PlugTVRestful. Data will be converted into information such as: time when the citizen leaves home and arrives at work associated to services of the city as traffic information, climate catastrophes and flooding situations. As we can see, the architecture of the platform is composed by 4 components:

1) *Architecture of the platform*

a) *Mini-PC (Gateway)* – A Mini-PC installed in the house of the citizen. It is connected to heterogeneous devices connected to the Internet and it is through this gateway that some of the data processing happen to detect patterns on the routine of the citizen;

b) *PlugTVRestful (WEB component)* – It is a Web component implemented as a service in the Java EE 7 platform and it is organized as a RESSTful service, which has the Framework Jersey as its base. Loggers of the devices are persisted and sent in Json format through this component;

c) *PlugTVMobile (Mobile component)* – It is the mobile component developed in the Android platform and it is installed in the Mini-PC attached to the TV via HDMI to collect the logger. With the citizen properly registered and authenticated in the system, the device receives permission to send data to the Web platform and this component sends a logger automatically every time the TV is turned on, in an automatic and transparent way to the citizen, collecting data such as: time the TV was turned on, house and city where the device is being monitored, latitude and longitude, information about the city such as Gross Domestic Product (GDP), current manager, revenues, expenditures and territorial area;

d) *Smart City Services* – Available services of the cities, such as:

- Transportation.
- Healthcare.
- Education.
- Public safety and security.
- Building management.
- City administration.

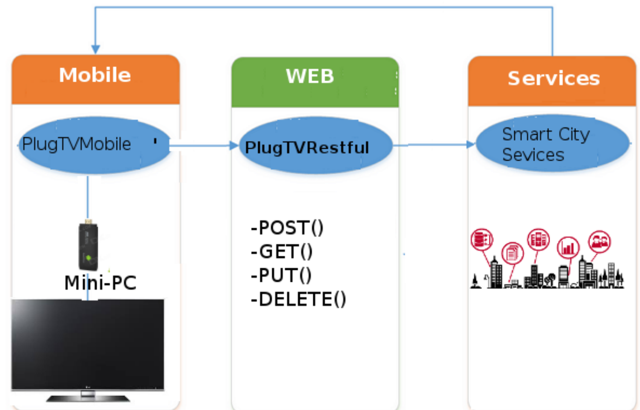

Figure 3. Architecture of the PlugTV.

B. Characteristics of the PlugTV

PlugTV is a proposal of developing software to collect and process data of the routine of the citizen through the TV. Because this a software, the development of the tool is based on some essential characteristics, such as: (a) reliability; (b) performance; (c) maintanability; (d) safety; (e) flexibility; (f) testing facility; (g) portability; (h) reusability; (i) interoperability.

Besides comprehending these characteristics, the PlugTV aims to eliminate two issues found in the development of applications for TV, which are: (a) dependency on the communication between the application and the Digital TV Middleware and how to order the application in the carrousel of data of the Digital TV providers; (b) existence of different platforms of application for Smart TV, since each manufacturer has their own SDK, which disables portability. With the use of PlugTV, these issues are abstracted, allowing an application to collect information through the TV and remaining in a RESTful server found in the clouds, without making these data go through the carrousel of data. PlugTV makes this application work in any Smart TV, regardless the producer and IPTV patterns.

Data will be collected every time the TV is turned on, in a transparent way for the user, so they can be analyzed and converted into information useful for the citizen, connecting them to services offered by the city, for example: traffic information, climate change and better use of resources, such as energy.

C. Data Structure of the PlugTV

PlugTV manages 2 types of data: house data, collected by devices (e.g., Smart TV) and settings data, which can be classified the following way:

1. **House settings:** Contains information about the house and has two entities: City Entity and House Entity;
2. **Device Settings:** Has two entities – Device and DeviceType. Device is the real device installed in the house and DeviceType classified the kind of device connected;

 ISBN: 978-1-61208-438-1

3. **Person Settings:** Has two entities: Citizen and Logger. Citizen represents all the citizen information and Logger contains information of all the devices of the citizen and routine data collected by the application.

Figure 4 illustrates the Diagram of Entity and Relationship of the PlugTV.

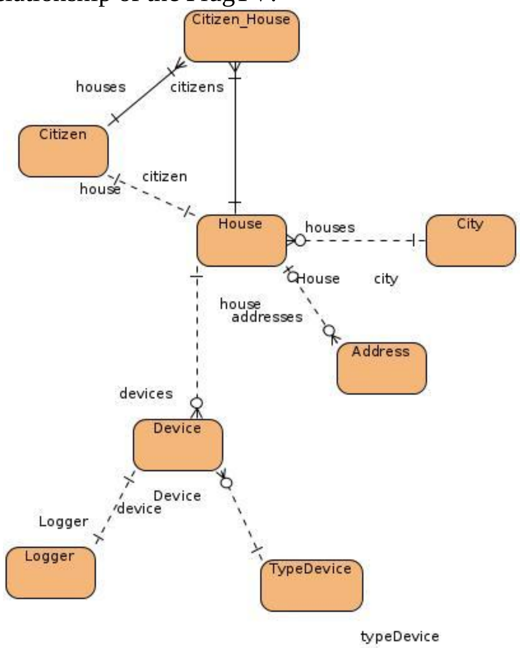

Figure 4. Design of the PlugTV database.

IV. TECHNOLOGIES APPLIED

PlugTV was developed in two modules: the one called ***PlugTVRestful*** was developed in the Java EE7 Platform and it is disposed as a ***RESTful*** server, with a Jersey Framework as base; the second module was implemented in the Android platform, running in a Mini PC attached to the TV. In order to manipulate the exchange of information with the server, the API JSON was used.

The main objective of the following sub-items is presenting artifacts generated during the implementation of the PlugTV.

A. Functional Requirements

The main functional requirements of the ***PlugTV*** are: (a) generating a routine logger of the citizen through the TV; (b) visualizing the collected data; (c) exporting the collected data in different formats, for example: xml, *json*, html, text/plain; (d) automatic saving of the TV logger.

The automatic saving of the TV log consists in handling data that will remain on the Restful server, called ***PlugTVRestful***. If the service that collects loggers is not available, data will be kept in the device to be synchronized after in the server.

B. Non-functional Requirements

Non-functional requirements refer to general features of the system, such as: safety, performance, distribution, maintainability and others. For the PlugTV, the following requirements were defined as non-functional: (a) Implementation using JAVA EE 7 to ensure portability; (b) Implementation using Android Version 4; (c) All API traffic should be over a secure connection using SSL; (d) Retrieved address and object information must be authenticated; Implementation of PlugTV Services

In order to provide services to be associated to services of smart cities, it was necessary to implement a services API of general use, including consultations to many data requests of the PlugTV. First, we are going to classify the types of service and their respective responses to requests. Services of the PlugTV were implemented as RESTful Web services and will be described after. Figure 5 presents the services and resources of the PlugTV.

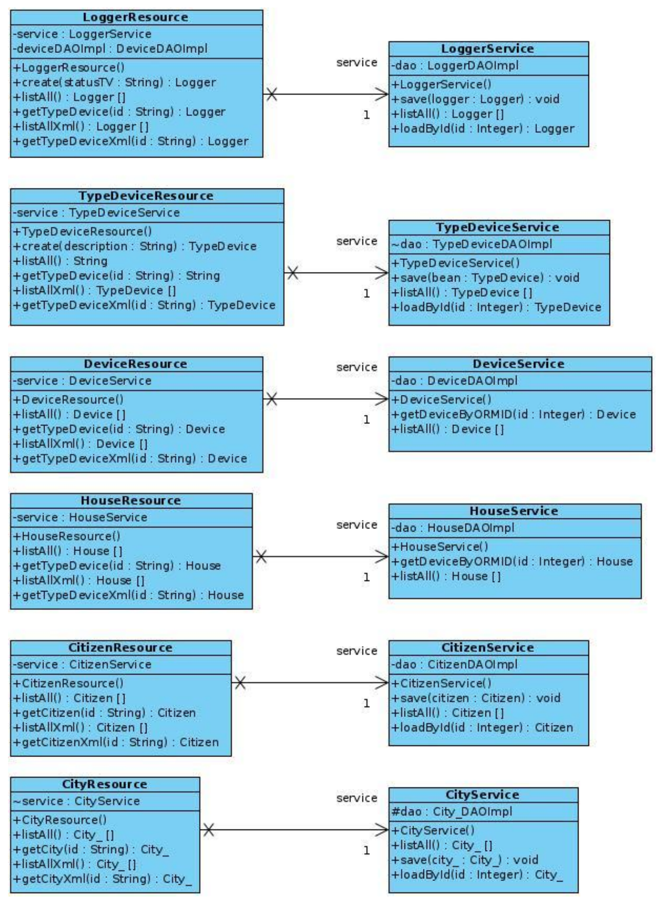

Figure 5. *PlugTV* services.

V. EXPERIMENTAL EVALUATION

This section discusses the method used to evaluate the proposed tool. In order to perform this analysis, two experiments were made, contextualized as tests. Two

 ISBN: 978-1-61208-438-1

servers were used to conduct the tests: the first representing the devices of the clients and the second representing the server, called **SERVER1** and **SERVER2** respectively.

SERVER1 was used to simulate users (TV) who send requests through the PlugTV tool. When it comes to the simulation of users, tests were recreated in the JMeter tool, version 2.3, a tool used to test the load in services offered by computer systems. This tool is part of the Jakarta project from the Apache Software Foundation. The computer called **SERVER2**, located in clouds, was responsible for hosting the web module **PlugTV** and process the requests sent by **SERVER1**.

Two tests called **Test1** and **Test2**, respectively, were performed to contextualize the experiments. The first simulated requests only where no event from the city was set by the tool, for example, a traffic jam at the time the citizen eventually leaves the house. In the second test, for every request made to the tool, an event of the city was associated. The metrics selected to track the performance of the tests were: *Times Over Time, Response Times Percentiles, Aggregate Graph.*

The setting of the requests of the users was made through Http Request the GET kind and configured as follows: Connect Timeout – 10.000(ms), Response Timeout = 10.000(ms). That is, if the request takes between zero and ten seconds to be opened or more than 10 seconds to be answered, this request is considered a defected one. The Path defined for the GET method described before follows the following format: /PlugTVRestful/logger/get/[request id]/.

A. *Times Over Time*

The objective of this metric is to measure the average time a request takes to be answered. That is, the time taken or the average delay between the start of a transaction and the results of it. These metrics are referenced and represented in JMeter by Times Over Time and Aggregate Graph.

In **Test1**, the user only sends his requests to module PlugTVRestful, which registers the logger of the TV. Since no event of the city was notified, an archive in the JSON format informs the logger was registered, unlike **Test2**. There, each request is associated to an event of the city, which besides sending the logger of the TV, an event associated to services of the city is signed, for example, traffic information, generating a JSON archive with information of the event.

In order to understand better the response time, the response time of each test is shown in Figure 6.

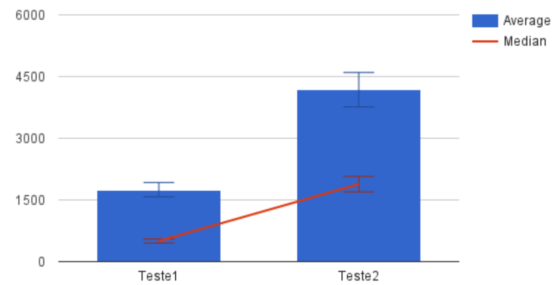

Figure 6. Response Time.

Percentiles are measures that divide the ordered sample (by ascending data order) in 100 parts, each one with a data percentage approximately equal. The k-nth Pk percentile is the x value (xk), which corresponds to the cumulative frequency of N k/100, where N is the sample size. Equation (1) demonstrates how the percentiles are calculated. The percentiles of the response time for each test are described in Figure 7. With an average response time of 1.749,4ms, Test1 is in a percentile of 79,5%. It means that 79,5% of the requests have a shorter or equal time than/to the average. Similarly, Test2 is in the percentile of 69,2%, with an average response time of 4.180,3 ms.

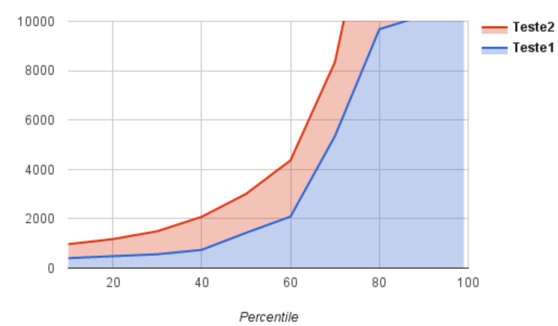

Figure 7. Percentile rank.

$$PR_x = \frac{\sum fb_x + \frac{f_x}{2}}{N}(100) \qquad (1)$$

B. *Success and Failure Rate*

The objective of this metric is to analyze the success and failure behavior in the requests measures in the tests.

In order to perform the tests in JMeter, a metric called Transactions per Second was used, where the requests were set to suppose there was a failure in case of the connection

 ISBN: 978-1-61208-438-1

takes more than 10 seconds to be opened or more than 10 seconds to be answered.

For Test1, 122,7 requests were successfully processed and 6,76 requests per second failed. For Test2, 114, 8 were successful and 23,20 failures, as the following figure 8 shows.

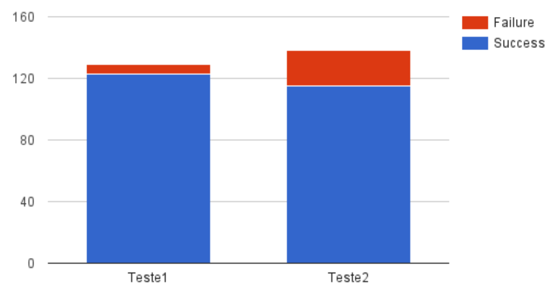

Figure 8. Success and Failure Rate.

VI. CONCLUSION

This paper proposed the development of a platform to integrate data of the urban space of a city with information on the routine of the citizen through a TV, in the context of the Internet of Things. The architecture is based on services and its objective is to provide information, abstracting the dependency of the communication of the TV with the Digital TV Middleware. For future works, the development and refining of the platform components, unit, integration and functional testing are expected. Another open question is increasing the number of devices connected to the tool, allowing a larger number of data. When it comes to data analysis, the development of a component that allows real-time analysis through techniques of Complex Events Processing (CEP) is desired.

REFERENCES

[1] J. Belissent, “Getting clever about smart cities: new opportunities require new business models,” 2010. [Online]. Available: https://www.forrester.com/Getting Clever About Smart Cities New Opportunities Require New Business Models/fulltext/-/E-res56701

[2] J. Jin, J. Gubbi, T. Luo, and M. Palaniswami, “Network architecture and QoS issues in the internet of things for a smart city,” 2012 Int. Symp. Commun. Inf. Technol., pp. 956–961, Oct. 2012.

[3] Washburn, D., Sindhu, U., Balaouras, S., Dines, R. A., Hayes, N. M., & Nelson, L. E., Helping CIOs Understand “Smart City” Initiatives: Defining the Smart City, Its Drivers, and the Role of the CIO.Cambridge, MA: 2010 Forrester Research, Inc. Available at http://public.dhe.ibm.com/partnerworld/pub/smb/smarterplanet/forr_help_cios_und_smart_city_initiatives.pdf.

[4] Dirks, S., Gurdgiev, C., & Keeling, M. (2010). Smarter Cities for Smarter Growth: How Cities Can Optimize Their Systems for the Talent-Based Economy. Somers, NY: IBM Global Business Services. Somers, NY: IBM Global Business Services. Available at ftp://public.dhe.ibm.com/common/ssi/ecm/en/gbe03348usen/GBE03348USEN.PDF

[5] Washburn, D., Sindhu, U., Balaouras, S., Dines, R. A., Hayes, N. M., & Nelson, L. E. (2010). Helping CIOs Understand "Smart City" Initiatives: Defining the Smart City, Its Drivers, and the Role of the CIO. Cambridge, MA: Forrester Research, Inc. Available from http://public.dhe.ibm.com/partnerworld/pub/smb/smarterplanet/forr_help_cios_und_smart_city_initiatives.pdf.

[6] S. Nanni and G. Mazzini, “Smart City, a model and an architecture of a real project: SensorNet,” in Software, Telecommunications and Computer Networks (SoftCOM), 2013 21st International Conference on, 2013, pp. 1–4

[7] F. G. Dohler M., Ratti C., Paraszczak J., “Mart ities,” no. June, pp. 70–71, 2013.

[8] S. Dirks and M. Keeling, “A vision of smarter cities,” New York IBM Glob. Serv., p. 18, 2009.

[9] G. Suciu, A. Vulpe, S. Halunga, O. Fratu, G. Todoran, and V. Suciu, “Smart Cities Built on Resilient Cloud Computing and Secure Internet of Things,” in Control Systems and Computer Science (CSCS), 2013 19th International Conference on, 2013, pp. 513–518.

[10] D. Jin, Y. Zheng, H. Zhu, D. M. Nicol, and L. Winterrowd, “Virtual Time Integration of Emulation and Parallel Simulation,” in Principles of Advanced and Distributed Simulation (PADS), 2012 ACM/IEEE/SCS 26th Workshop on, 2012, pp. 201–210.

[11] E. Theodoridis, G. Mylonas, and I. Chatzigiannakis, “Developing an IoT Smart City framework,” in Information, Intelligence, Systems and Applications (IISA), 2013 Fourth International Conference on, 2013, pp. 1–6.

[12] CISCO, The Internet of Things, Infographic, available online at http://blogs.cisco.com/news/the-internet-of-things-infographic, 2011

[13] E. Aubry, T. Silverston, A. Lahmadi, and O. Festor, “CrowdOut: A mobile crowdsourcing service for road safety in digital cities,” in Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on, 2014, pp. 86–91.

[14] Waze (Community-Based Traffic and Navigation Application). Retrieved June 29, 2015 from https://www.waze.com.

[15] M. Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

 ISBN: 978-1-61208-438-1