Testing Smart Cities Through an Extensible Testbed

A Testbed Framework For Smart Cities Validations

Guilherme Luiz Mario de Medeiros[1], Felipe Silva Ferraz[1,2], Gutemberg Rodrigues Costa Cavalcante[1]

[1]CESAR
Recife Center for Advanced Studies and System
Recife, Brazil
guicaraciolo@gmail.com
fsf@cesar.org.br
gutembergrcc@gmail.com

[2]Informatics Center
Federal University of Pernambuco
Recife, Brazil
fsf3@cin.ufpe.br

***Abstract*—Urban areas around the world are being crowded and, in some cases, over populated. This leads to a new set of modern life problems, like traffic jams and natural resource depletion. Information technologies, under the alias of smart city or Internet of things, play a very important role helping analyze, understand, and solve these problems. The present work discusses the need for testing on smart cities and Internet of things field of study. Following this line of thought, it goes through how other researchers are testing and validating their projects. Finally, it presents an easy to use and customize way of generating cheap data mass for project validation.**

Keywords-Testbed; Data Mass; Smart Cities; Internet of Things

I. Introduction

The growing number of citizens in urban areas is creating a new range of modern problems to humankind. Resource distribution, government transparency, security, mobility, and life quality are just a small set of these new challenges [1]. Some of these problems can be seen worldwide. For example, it is easy to think of at least one name of a city suffering from traffic jams [3], which is one of many mobility problems.

In addition, there is a set of invisible problems related to the growth of population on Earth. Urgent problems like pollution and the depletion of natural resources are already a huge concern in the modern society. The number of people in cities, and how those people behave, affects not only that small location on Earth, but also the entire planet [3][4].

Modern human life has a helpful ally on fighting all of those problems. The increasing power of computational devices and the amount of data gathered about, almost, everything, can help researchers and scientists understand how those problems appear and behave and simulate tools and scenarios, trying to find ways to solve or prevent those problems [4].

The main problem of computational tools is the need to test it. Researchers need to ensure their tools are working as expected, under any circumstance. In Computer Science and software engineering, there are different concepts and tools to test hardware and software, which are being used worldwide by the tech industry. Relying on those tools and techniques would give researchers the certainty their solution may be on the right tracks [5][6].

There are different ways to ensure that software works as expected. One of them is by using a testbed. A testbed comprehends a huge set of tools. In this set, there are tools that behave as a real environment, where other software can be executed in it, “thinking” they are executing on the real environment. In addition, in this set of tools, there are data generators, which simulate a real environment, as the previous testbed, but instead of executing the test subject in it, it outputs a dataset from the simulated environment, which can be used as input for the software the researcher wants to test [5][6][7].

The present work introduces a data simulation testbed, which can simulate the desired target area, and return an almost unique data set. More than that, the final product of this research is a framework to construct testbeds for data mass generation to help testing smart cities and internet of things applications. The main reason of this research is to provide to other researchers a simple to use and extend tool.

The current paper is divided on the following sections: Section II describes the concept of a smart city and what researchers are doing with it; Section III shows how other researchers are trying to solve the lack of smart cities testing tools; Section IV presents how the proposed framework was constructed, the idea and architectural decisions behind it’s functionalities; Section V displays the framework validation; Section VI holds a final discussion about this work.

II. Smart Cities

The expansion of urban areas and the growing number of people on those spaces are creating new problems to humankind like resource distribution or service management. Solving these problems is not simple, which is leading scientists into looking for solutions on, but not exclusively, smart cities [8].

The concept of smart city may vary by researcher, university, or book, but the main idea is the same: try to solve urban problems, save governments money and improve life quality by using Information Technologies (software, hardware, networks, and sensors) [9][14]. However, like any other research field, smart city also needs validation. For this, the best way of validation is gathering real data from urban area and replicate it to test the solution. The main problem is gathering real world information, which can be expensive, for not mentioning administrative barriers. This can delay the development of a good idea for a long time. In addition, after gathering the desired data, it can disprove the project; meaning money and time were wasted. By exposing this thought, it is easy to understand the need for a reliable and cheap data mass for initial validations on smart cities projects to avoid time and money being wasted [9][10].

Nowadays, it is starting to be common to find public data masses and Application Program Interfaces (APIs), provided by governments for researchers. Nevertheless, those data does not show how citizens behave. For security and bureaucratic reasons, most public data masses relate to government controlled resources usage. Even though these data sets can be helpful and insightful, they are not suitable for all researches [9][10].

To circumvent those problems, researchers are developing testbeds based on real data and observations to generate good data masses for project validations. This may not represent the real world behavior, but such low cost approach is helping researchers to test and validate their ideas early on initial stages.

III. RELATED WORK

In this section, there are enumerated some works and researches that identified the same problem, by quickly explaining their approach to solve it.

A. Simulating Smart Cities with Deus

The study introduces a testbed able to create data mass to test wireless communications infrastructure in a smart city. Its data mass is generated by the frequency probability of a series of discrete events and distributing those events along a time period, using a deterministic algorithm. In this testbed, a discrete event is any state change or communication between nodes – which can be citizens or sensors. The proposed testbed can generate a high amount of data in a small period. On the other hand, it does not offer easy ways to configure data generation [11].

B. Smarty City Application Testbed

This work proposes a testbed for data generation through simulating citizens and resource usage. The application developed provides a way to input configuration options and the testbed tries to simulate behaviors and generate data based on the user input. It seems to be a good idea to enable the test of specific scenarios. The proposal cannot be extended, which means other researchers will not be able to add more modules or entities to the tool simulation, being limited to its generation capabilities [7].

C. Smart City – Platform For emergent Phenomena Power System Testbed Simulator

Differently from the two previous testbeds, this proposal is not about generating data to validate smart cities solutions. The paper proposal is to create a virtual environment where Power Grids (Electrical Grids) projects for smart cities projects can be tested. The main idea is to develop the desired solution in this simulator; while it generates usage patterns and unpredictable events (like natural disasters). This represents a good approach for specific problems especially smart systems that may need to take quick measures without human intervention [5].

D. SmartSantander: The Meeting Point Between Future Internet Research and Experimentation and the Smart Cities

Smart Santander [6][12][15] is a testbed for smart city projects running in a real live city. Researchers can implement their projects and analyze how it would behave in the real world. The architecture is based on different Tiers, and the communication between components among them. Each Tier communicates with a capillary network that carries information to application servers. To facilitate the implementation of new solutions, this architecture is based on only four subsystems, with simple interfaces to use and access.

This project is a major milestone in the field of smart cities. It is hard to point downsides on the proposal since they can also represent new reasons for studies, as the scenario is the most suitable for such.

E. IoT Testbed Business Model

The study presented by Silva et al. [15] is a business model for internet of things testbed deployment on a real environment. It is a derivation of the business ideas and commercialization principles behind Smart Santander [6][12][15]. The model considers four main aspects of a business: infrastructure; value proposition; customers; and financials. Finally, it shows how to detect key partners, key activities, cost structures, and revenue streams, so governments, private corporates or universities can try to implement their own “Smart Santander”.

IV. FRAMEWORK FOR DATA GENERATION

The final product of the present paper is a framework to enable other researchers to create their own testbeds. Those testbeds will generate data mass by simulating a virtual city. For this to happen, the study considered two different configuration moments: simulator configuration and execution profiles. The simulator configuration deals with what the final testbed will simulate and is achieved by adding, removing, or modifying simulation modules. Researchers can use those modules to enable different behaviours to their simulated city. On the other hand, the execution profiles modifies how the simulator will behave. While the simulator just shapes a city, the execution profiles describes how much resources the city will consume during its simulation, which can modify all modules behaviours.

The proposed framework is based on three different parts (as shown on Figure 1): data storage, a Web based Guide User Interface (GUI) and the core application. After generating data, the framework produces a single JavaScript Object Notation (JSON) file, which can be used to validate any smart city project.

A. Data Storage

The main goal of the data storage is to hold execution variables and information, removing this responsibility from the core application. This means the testbed being developed does not need to deal with memory swap, indexing or language garbage collector. Also, abbiding to this architecture decision enables the framework to outsource this concern to another process or software, parallelizing memory management and input and output blocking operations.

For this task, the framework relies on MongoDB - a cross platform document-oriented database, adopted by a number of the most popular websites and services. It focuses on giving fast reading and writing operations. To achieve this, it indexes data into hashes on memory. On the other hand, it does not provide the most common relational operations, like JOIN statements or transactions.

B. Web GUI

The core part of the proposed framework can accept user input, as execution profiles, to generate data mass for specific scenarios. To enable this, the framework provides a stand-alone application, which can behave like a HTTP Web server. Then, the application can connect to the Data Storage (previously described), to save new execution profiles or to download JSON files generated by previous executions. To be able to access all of its functionalities, the user just need to connect to it with a common Web Browser.

C. Core Application

The core application is subdivided in two parts: the Execution Manager and the Simulator. Figure 2 shows how the Core Application and all of its parts behave.

1) Execution Manager

The Execution Manager is capable of connecting to the Data Storage, verify if there are execution profiles to simulate and, for each one of them, raise a new instance of the Simulator. In addition, this Manager holds the responsibility to generate JSON files at the end of each simulation execution.

2) Simulator

The simulator exposes two methods for developers, and comes bundled with a set of simple basic modules.

a) Initialization Method

This method exists for importing and initializing user created and basic modules into the Simulator. By default, it imports and initializes all the basic modules that came bundled with the final application.

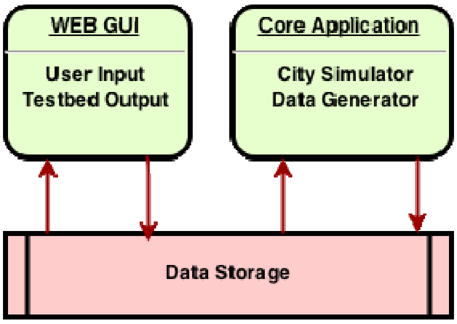

Figure 1. Three Parts Architecture.

This method receives as input the current execution profile, which should be used to initialize the modules that execution method will execute.

b) Execution Method

Three tick events compose the execution method: day changed; hour changed; minute changed. Developers must use at least one of these events to configure and execute these simulation modules. Since the entire execution environment exists on each of these events, everything can be used as input parameter for each module execution method. By doing this, it is easy to inject the current execution date time, or even all the existing citizens, into the module execution.

The goal of splitting the Execution Method into tick events is to facilitate the identification of sets of actions each module may need to perform according to the time of the day. More than that, it helps overhead reduction by avoiding unnecessary method calls. For example, the “traffic module” needs to be simulated each minute, so it can generate data as real as possible. On the other hand, “education access module” can be simulated once a day, may be at the end of the day (day changed event), since it may only need to simulate students score based on its frequency. In addition, adopting this architecture enables modules to have different execution methods for each tick event, improving readability on module’s code.

Notice that, like the real world, time is an incremental value, being orchestrated by the smallest time unit, which is, for the proposed framework architecture, the minute unit. The default real world, and framework, behavior is always increment the smallest unit by one. For the framework, the incremental value can be modified (in the Initialization Method) to make the simulation better suit the target project needs. For example, instead of incrementing by the default value, it could increment the minute unit by ten, reducing the number of minute tick events, decreasing generation time.

Finally, this architecture opted for not using parallel module execution, like threads or sub-process. This decision relates to the passage of time and each module execution state. Using threads or sub-process would force the framework and its modules to implement execution signals to ensure that different modules are still executing the same simulation moment. In addition, since modules can be used as parameters for other modules executions, using parallel processing could damage this concept, since injected modules could change their own execution state in the middle of the execution of the dependant module, making the chain execution fail.

c) *Default Basic Modules*

The framework comes bundled with three basic modules, which represents basic components of a city. Even though those modules can be used to generate data, their only purpose is to serve for this paper validations.

The bundled modules are:

- The City Manager – capable of generating a city. The size of the output city is based on user input (from the WEB GUI). The final city is randomly generated based on a predefine probability for each created block: 65% chance to be a home block; 25% chance to be a workplace; 1% chance to be a hospital; 1% chance to be a police station; 3% chance to be a school; 5% chance to be a leisure block and 1% chance to be a university block.
- The Citizen Manager – generates citizens for the simulation, gives those homes and jobs, and makes them walk through the city. Jobs and age distribution is based on a set of predefined “citizens profile”, where each profile has a chance of service consumption. The Citizen Manager tries to distribute this profiles based on input for service access.
- The Transport Manager – can only be initialized by injecting a city into it. At each iteration, receives a list of citizens that want to move to another location, tries to move them, and remove from this list the citizens that got to their destination.

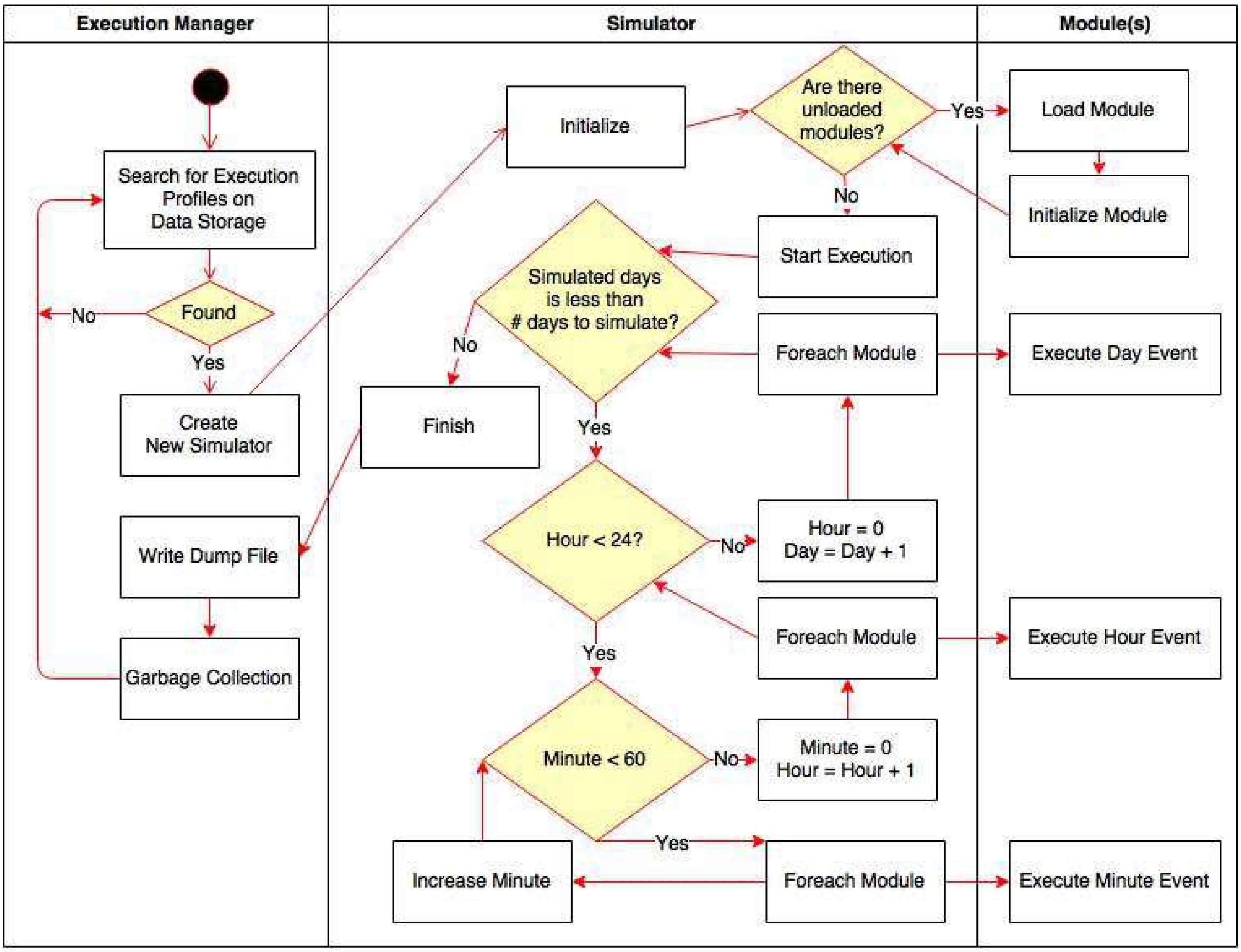

Figure 2. Core Application Workflow.

V. Validation

To validate the framework, three different scenarios were created: a simple village; *Espinheiro*; and *Boa Viagem*. While the village scenario tries to represent a fictional small village, which will help to validate the framework speed for small inputs, both *Espinheiro* and *Boa Viagem* scenarios try to mimic two huge neighborhoods from *Recife* (a city in Brazil). For validation and testing, data about those neighborhoods were gathered from [13]. Since this data source only deals with area size and number of citizens, and the framework needs to have an execution profile in order to work, for service consumption behaviour, random numbers were used. Even though these numbers may not represent real world services usage, they are good enough to validate the framework. The input data for each execution scenario can be found in Table 1. For these validations, the framework's default modules were used. Because of that, since the city generation module, created to validate the framework, only deals with blocks of buildings, for *Espinheiro* and *Boa Viagem* scenarios, a block was considered as having 100m2. Finally, each scenario was executed three times. A summary of each execution can be found on Table 2.

TABLE I. VALIDATION SCENARIOS

Input	Scenarios		
	Village	***Espinheiro***	***Boa Viagem***
City Width	10	72	251
City Height	10	100	251
N. of Citzens	2000	10000	130000
N. of Days to Simulate	90	90	90
Total Education Service	35000	30000	130000
Total Health Service	41000	80000	877500
Total Transport Service	280000	2250000	12090000

To execute these validations, a Virtual Private Server (VPS) were used, with the following environment specifications:

- One Virtual CPU with 2GHz.
- 1 GB RAM
- 30 GB SSD Hard Drive
- Ubuntu 14.04x64
- Python 2.7.6
- MongoDB 3.0.2.

Analyzing the summary output, it is easy to understand the impact the execution profile has on execution time and generated file size. Since for each scenario the input grew on city size, number of citizens and service consumption, the output for them also grew on execution time and generated file size. This means that a testbed generated by the framework is reusable for different city behaviours in the same virtual city, which means one of the two configuration moments, described in section IV, was achieved.

TABLE II. EXECUTION OUTPUT

Output	Village			Espinheiro			Boa Viagem		
	First	***Second***	***Third***	***First***	***Second***	***Third***	***First***	***Second***	***Third***
Execution Time (Seconds)	865.26	832.18	813.12	5479.79	5798.20	5272.86	32308.51	32420.90	32511.08
Data Mass File Size (MB)	63.8	60	59.2	397	396	387	1.7 GB	1.7 GB	1.7 GB
Home Blocks	73	57	66	4560	4636	4648	40826	40766	40802
Work Places Blocks	18	26	20	1857	1785	1822	15603	15704	15651
Hospital Blocks	1	2	1	68	87	62	617	637	694
School Blocks	5	6	2	220	217	210	1946	1813	1976
University Blocks	1	1	2	39	40	30	303	285	301
Police Station Blocks	1	2	1	74	70	56	643	646	647
Leisure Blocks	1	6	8	382	365	372	3063	3150	3109
School Citzens	72	94	103	851	849	831	366	364	364
Graduate Citzens	110	104	102	831	822	854	344	382	382
Gratuate-worker Citzens	112	88	87	826	819	785	356	349	349
Postgrad-worker Citzens	95	103	97	826	844	864	379	350	350
Worker Citzens	955	972	979	6666	6666	6666	64631	64667	64667
Retired Citzens	656	639	632	0	0	0	63924	63888	63888
Avarage Citzens Age	50	49	49	35	35	35	58	58	58

Continuing with the summary analysis, it is also clear how biased is the default module for city generation. Previously, it was mentioned that the default module has a probability for each block creation. Taking the "police station" and "hospital" probability as example, which is a 1% chance, and looking at the results, we can visualize the number of blocks for theses block types is always near 1%. Even though their coordinates may be very different, which may affect traffic related studies, the way blocks are being generated for different size of cities may not represent a city growth behavior.

In addition, the chosen framework architecture may not suit for larger simulations, since it does not deal with parallelism. The effect of this decision can be seen on the execution times presented in Table 2. The smaller scenario took fourteen minutes to process and the biggest scenario took eight hours. However, the proportion between each configuration entry for their execution profiles is not higher than sixty times. Even though execution time may be reduced by using a powerful CPU and more RAM memory, the gain may not be perceived on large simulation scenarios.

Finally, these validations prove that the framework really works, that it can be customized by different service modules, and also show how the execution profiles may affect the execution time and final file size. Even though the default modules work well and may be used for some use cases, they only exist to show how modules work inside the proposed framework and, as this validation shows, it is highly recommended that researchers extend or replace the default modules, or even add more city service modules.

VI. CONCLUSION

The effort of testing a solution is compensated by finding problems as soon as possible. Finding problems on tools in the early stages of a development, means fixing the solution early, when it is easier and cheaper. Also, battle testing the software or hardware being developed reduces the chance of an unknown problem being released worldwide. Even though this is common in the tech industry, the reasoning of using tests can be easily related to Smarty Cities and Internet of Things research.

The result of this paper is a framework for data generation through cities simulation. The generated data can be used in any smart cities or internet of things research. In addition, the proposed tool means a cost reduction, since researchers do not need to waste money on gathering real environment data until it is necessary. As shown in the validation section, researchers can simulate ninety days in only eight hours, which is a precious time saving. However, this time could be decreased if the framework supported parallel processing.

For future works, there is a need to research more on the framework modules. Implementing new modules for specific set of city and world functionalities would be helpful for other researchers. In addition, in the software engineering field, there is a need to find ways of optimizing the time the framework takes to generate data, but keeping it easy to use and customize. It would be great to see researches and projects on Smart Cities and Internet of Things using the testbed framework as one of its tools for project test and validation.

REFERENCES

[1] C. Harrison et al., "Foundations for Smarter Cities," IBM J. Res. and Dev., Jul. 2010, pp. 1-16, ISSN: 0018-8646.

[2] G. Coulson et al., "Flexible Experimentation in Wireless Sensor Networks", in Magazine Communications of the ACM, Volume 55 Issue 1, Jan. 2012, pp. 82-90, doi: 10.1145/2063176.2063198

[3] S. Dirks and M. Keeling, "A vision of smarter cities: How cities can lead the way into a prosperous and sustainable future," IBM Inst. Bus. Value, in Executive Report, June, 2009.

[4] Forrester. *Helping CIOs Understand "Smart City" Initiatives*. [Online]. Available from http://www-935.ibm.com/services/us/cio/pdf/forrester_help_cios_smart_city.pdf 2015.10.28

[5] L. Lugaric, G. S. Member, S. Krajcar, and Z. Simic, "Smart City - Platform for Emergent Phenomena Power System Testbed Simulator," IEEE, Innovative Smart Grid Technologies Conference Europe (ISTG Europe), Oct. 2010, pp. 1–7, doi:10.1109/ISGTEUROPE.2010.5638890.

[6] D. Carboni, A. Pintus, A. Piras, A. Serra, A. Badii, and M. Tiemann, "Scripting a Smart City: The CityScripts Experiment in Santander," 2013 27th Int. Conf. Adv. Inf. Netw. Appl. Work., IEEE, Mar. 2013, pp. 1265–1270, doi:10.1109/WAINA.2013.85.

[7] D. Silva, F. Ferraz, and C. Ferraz, "Smart City Applications TestBed Towards a service based TestBed for smart cities applications," in SOFTENG 2015 : The First International Conference on Advances and Trends in Software Engineering Information, Apr. 2015, pp. 104–107, ISBN: 978-1-61208-449-7.

[8] F. Ferraz, C. Sampaio, and C. Ferraz, "Towards a Smart City Security Model Exploring Smart Cities Elements Based on Nowadays Solutions," ICSEA 2013, The Eight International Conference on Software Engineering Advances, 2013, pp. 546–550.

[9] T. Nam and T. a. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions," Proc. 12th Annu. Int. Digit. Gov. Res. Conf. Digit. Gov. Innov. Challenging Times - dg.o '11, Jun. 2011, pp. 282, doi: 10.1145/2037556.2037602.

[10] W. Da Silva el al., "Smart cities software architectures", in Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13, 2013, pp. 1722, doi: 10.1145/2480362.2480688.

[11] M. Picone, M. Amoretti, and F. Zanichelli, "Simulating Smart Cities with DEUS," Proc. Fifth Int. Conf. Simul. Tools Tech., 2012, pp. 172-177, ISBN: 978-1-4503-1510-4.

[12] L. Sanchez et al., "SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities," Future Network and Mobile Summit (FutureNetw), Jun. 2011, pp. 15-17, ISBN: 978-1-4577-0928-9.

[13] Recife Open Data Website. [Online] Available from: http://dados.recife.pe.gov.br/ 2015.09.30

[14] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," IEEE Internet of Things Journal, Vol. 1, No. 1, Feb. 2014, pp. 22-32

[15] E. Silva and P. Maló, "IoT Testbed Business Model," Advances in Internet of Things, 4, 2014, pp. 37-45, doi:10.4236/ait.2014.44006