

Modeling and Simulation of a Smart Street Lighting System

Paulo G. Teixeira¹, Bruno G. A. Lebttag¹, Luma W. de Oliveira¹
Sérgio T. de Carvalho¹, Ernesto F. Veiga², Cleomar de S. Rocha^{1,2}

¹Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 131 – CEP 74001-970 – Goiânia – GO – Brazil

²Secretaria de Ciência, Tecnologia e Inovação – Prefeitura Municipal de Aparecida de Goiânia
CEP 74968-500 – Aparecida de Goiânia – GO – Brazil

{paulogabriel, brunogabriel, lumawanderley, sergio@inf.ufg.br}

{ernesto.veiga, cleomar.rocha@aparecida.go.gov}

Abstract. *Smart cities comprise highly dynamic, complex and software-intensive systems planned and created to cope with problems inherent to the increasing world population, such as need for life quality in cities, intense traffic, and sustainability. Owing to such complexity and dynamics, static notations, such as Unified Modeling Language (UML) and Systems Modeling Language (SysML), are often unable to support a precise conception, planning and design of those smart cities. In turn, the combination of traditional modeling and simulation (M&S) could leverage the design of such systems by enabling the analysis of both structure and behavior aspects still at design-time. The main contribution of this paper is offering evidence to support the premise of M&S be a promising paradigm for software engineering, including in smart cities domain. We report findings on a study carried out for assessing, via M&S, two architectures for a Public Street Lighting System (PSLS) of a smart city. The first architecture uses a hub as mediator between luminaries and PSLS. The second architecture uses software-intensive luminaries connected to the PSLS using publish/subscribe architectural style. Preliminary results show that the publish/subscribe architecture delivers a greater degree of correctness than the former. Additionally, we conjecture that the adoption of M&S can foster the identification of smart cities behaviors and the identification of, at design-time, properties that could be noticed only after the system has been already deployed.*

1. Introduction

The world population is expected to reach 8.5 billion by 2030. Currently, half of humanity lives in cities [Khatoun and Zeadally 2016]. The accelerated development of cities brings several problems related to the unplanned growth of urban areas [Carvalho et al. 2018]. Consequently, innovative technologies should be created to guarantee a better quality of life for population and optimize the management of resources and infrastructure [Pla-Castells et al. 2015]. Smart cities depend on advanced data processing with the objective of contributing to the social, environmental and economic needs of society [Yin et al. 2015]. Software solutions can optimize city operations and services through connecting citizens and providing comfort and security [Kumar et al. 2018]. However, modeling and developing software for smart cities face additional challenges, since smart cities are (i) complex systems and (ii) highly dynamic [Picone et al. 2012,

Manzano et al. 2019], i.e., the entire city architecture continuously change due to systems that join and leave the system during its operation, such as autonomous cars that join or leave the traffic system, or mobiles and drones that can be turned on or off, joining and leaving the city system.

To deal with such complexity and dynamics, models are essential. By means of abstraction, models enable the engineer to focus on the important aspects of a system [Selic 2003]. However, current practices of software engineering recommend the use of UML and SysML, which present important drawbacks in regards to represent of internal structure and the system's behavior, as well as in the interoperability with other systems [Debbabi et al. 2010]. Those notations are thus unable to represent the characteristics of smart cities systems, such as interoperability between independent systems, sustainability, real-time monitoring, historical data, mobility, availability, privacy, sensors and distributed processing, service composition and integrated urban management, social aspects, flexibility and extensibility [da Silva et al. 2013]. On this direction, simulation has been successfully used in several disciplines, such as economy, biology, social sciences, and also in software engineering, to support the visualization of the dynamic structure and behaviors of systems. Simulations can anticipate, at design time, failures and behaviors that could potentially occur at run time [França and Travassos 2013, Manzano et al. 2019].

The main contribution of this paper is providing evidence of the important advances that the interplay between modeling and simulation (M&S) can bring to software engineering. By using M&S, we performed modeling, simulation, and comparison of two architectural alternatives for a software-intensive subsystem of a smart city: a Public Street Lighting System (PSLS). This study was developed in partnership between a university and a prefecture technology department. The aim of this research was to investigate whether the use of simulation could provide insights about the structure and behavior of this software-based system. By observing its behavior in two architectural alternatives, the stakeholders could decide on the better solution for being implemented and deployed in a real smart city project being carried out still at design-time. Preliminary results reveal that a publish/subscribe architecture was able to deliver all the expected results. The remainder of the paper is organized as follows: Section 2 briefly introduces the foundations of our research. Section 3 presents the Public Street Lighting System (PSLS). Section 4 details the research methodology and experimental design of our study. Section 5 discusses the results. Finally, Section 6 brings the conclusions and future work.

2. Public Street Lighting System

The PSLS is part of a smart city project currently developed as an innovation project of Aparecida de Goiânia city, in Brazil. PSLS project seeks, through systems interoperability, to support strategic actions to be undertaken depending on events that occur in the city. The improvement of public lighting network through the implementation of an automated operation gives the system conditions to be an important social management and smart city mechanism. The addition of software in PSLS could bring the following advantages and functional possibilities: (i) reduction in energy consumption, (ii) changing light colors in case of accidents to notify drivers; and (iii) festive occasions, showing country flag colors in lights.

The project for the PSLS development involves: 1) software creation; 2) the implementation of this software in specific hardware; and 3) hardware deployment in the environment, distributed geographically in the city area, aiming at (i) greater security and appreciation of local historical and cultural heritage; (ii) to make the system own energy consumption more efficient and sustainable; (iii) real-time event monitoring and take immediate action; (iv) control of lighting intensity in municipal public roads; (v) measuring and storing information on actual energy consumption; and, (vi) to record the behavior changes of the components, making them available in real time.

3. Simulation Model Description

The *in-silico* experiment was conducted according to França and Travassos guidelines [de França and Travassos 2016]. Our study consists of simulating part of an Smart Public Lighting System (PSLS), responsible for changing the behavior of the luminaires located in the vicinity of traffic accidents, thus providing an alert zone for drivers traveling through this region, avoiding further accidents. We modelled a hub architecture initially proposed by the secretariat of innovation and technology of Aparecida de Goiânia, hence from this initial modeling was perceived some possible improvements. To have a second architecture as comparison, we modeled an architecture using the publish/subscribe architecture that could also meet the system needs. We validated both architectures with the prefecture team and specialists.

The objective of this study is to evaluate the functional suitability of two proposed architectures, specifically dealing with functional correctness, which comprises the degree to which a product or system provides the correct results with the degree of precision required [ISO/IEC 2010]. We establish the following research question (RQ):

RQ: *What is the degree of functional correctness of each of the architectures?*

Rationale: this question evaluates whether the proposed architectures are able to meet the needs and requirements established by technology department of the Aparecida de Goiânia city, demonstrating that the system behaves and generates the results desired.

Metric: *functional correctness*: measured by the relation between the number of luminaires that changed their behavior due to an accident, and the total number of luminaires that should have changed their behavior within the warning area.

The DEVS (Discrete Event System Specification) formalism was used, we also used the MS4ME¹ M&S environment, which offers a restricted English interface to generate models and graphically simulate systems behaviors and structure. The simulation took for 1 min and 54 seg in Processor Intel Core i7-5500U CPU @ 2.40GHz, with 8 GB of RAM Memory, HD of 500 GB, and running Ubuntu 18.04.2 LTS 64 bits. During the simulation, all data will be sent to the OCC, and were stored in text files. Based on these data, we checked the total number of luminaries that changed their behaviors comparing them to the expected number according to the established criteria of 150 meters alert zone in order to assess the functional correctness.

Results. We selected 15 points among 45 accident points located in the city, where 5 of them are the ones covered by cameras, to create the figure 1. From the analysis of

¹<http://www.ms4systems.com/pages/ms4me.php>

the results obtained through the simulation related to this points, the chart shows the comparison between the quantity of luminaires that should have changed their behaviors to form the warning zone of 150 meters around the accident point and the total quantity of luminaires that really changed their behavior in the simulation of both architectures in the occurrence of a reported car accident. In some accident points, no luminaires will be able to change their behavior due to the fact they are far from the 150 meters of distance from the accident point and so the expected number of luminaires will be 0, which is the case in the points 3 and 5. Besides that, in nine points, both architectures were able to change their behavior according to the expected result. In 4 points the hub architecture was not able to change the behavior of all luminaires as it was expected. We calculated the functional correctness from the obtained results of all the accident points against the expected results. The hub architecture obtained 75.69% of functional correctness, due to the fact it failed to change the behavior of some luminaires in 12 accident points. The publish/subscribe architecture obtained 100% functional correctness, it was able to change the behavior of the luminaires of all accident points.

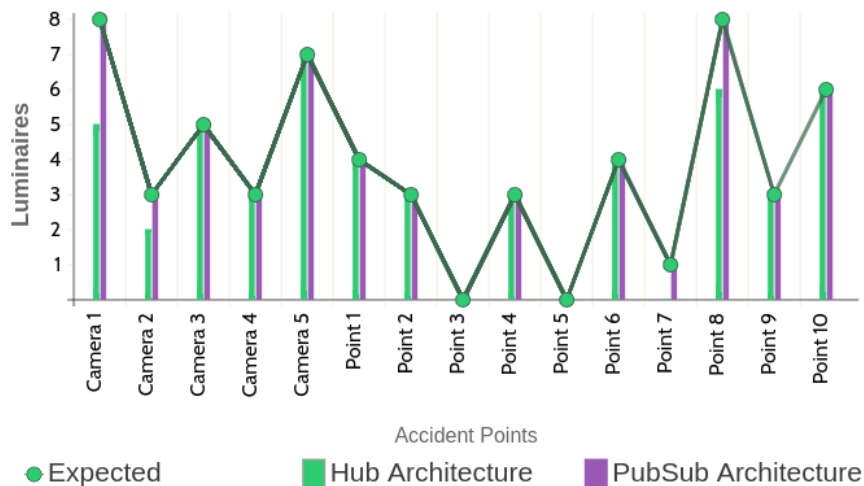


Figure 1. Comparison of the number of luminaires that changed their behavior in each architecture

4. Conclusion and Future Work

This paper presented a modeling and simulation of a Public Street Lighting System (PSLS), which intends to use software to change the behavior of public luminaires as a result of critical events that can bring risks to the population, such as car accidents. It was proposed two possible architectures to the PSLS system: one of them using the publish/subscribe architectural pattern where the luminaires connect directly to the PSLS and the other using hubs as intermediates between the luminaires and the PSLS. Both architectures were implemented using DEVS to simulate and evaluate the functional correctness of each one. The simulation results have shown that the publish/subscribe architecture was able to deliver all the expected results in addition to obtain more availability and easiness in its implementation whereas the hub architecture had problems related to the distance of the accident point in relation to the its cover radius, furthermore it can constitute a problem because it will be unique point of failure. the hub architecture can also increase the manual labor due to its implementation difficulties.

As future work, we intend to model and simulate other PSLs functionalities, such as storing and measuring information about the real energy consumption to allow to trigger the team in the field to fix possible incidents or problems, updating the OCC about the situation status. Other possible future work is to alter the system to deal with the traffic light systems, allowing it to integrate the systems together in order to improve car traffic when there is the occurrence of a car accident.

5. Acknowledgements

We thank CNPq for partially supporting this research under grants number 130337/2019-6 (first author). We also thank CAPES for partially supporting this research under grants numbers 1818312 (second author), 1836278 (third author).

References

- Carvalho, G. A., Neto, P. P. B., Cacho, N., Adachi, E., and Lopes, F. (2018). Plataforma rota: Histórico, desafios e soluções para segurança pública em cidades inteligentes. In *1º WBCI 2018*, number 1/2018. SBC.
- da Silva, W. M., Alvaro, A., Tomas, G. H., Afonso, R. A., Dias, K. L., and Garcia, V. C. (2013). Smart cities software architectures: a survey. In *SAC*, pages 1722–1727. ACM.
- de França, B. B. N. and Travassos, G. H. (2016). Experimentation with dynamic simulation models in software engineering: planning and reporting guidelines. *Empirical Software Engineering*, 21(3):1302–1345.
- Debbabi, M., Hassaine, F., Jarraya, Y., Soeanu, A., and Alawneh, L. (2010). *Verification and validation in systems engineering: assessing UML/SysML design models*. Springer Science & Business Media.
- França, B. B. N. d. and Travassos, G. H. (2013). Are we prepared for simulation based studies in software engineering yet? *CLEI electronic journal*, 16(1):9–9.
- ISO/IEC (2010). ISO/IEC 25010 System and software quality models. Technical report.
- Khatoun, R. and Zeadally, S. (2016). Smart cities: concepts, architectures, research opportunities. *Commun. Acn*, 59(8):46–57.
- Kumar, N. M., Goel, S., and Mallick, P. K. (2018). Smart cities in india: Features, policies, current status, and challenges. In *2018 ICSESP*, pages 1–4. IEEE.
- Manzano, W., Graciano Neto, V. V., and Nakagawa, E. Y. (2019). Dynamic-sos: An approach for the simulation of systems-of-systems dynamic architectures. *The Computer Journal*.
- Picone, M., Amoretti, M., and Zanichelli, F. (2012). Simulating smart cities with deus. In *Proceedings of the 5th International ICST*, pages 172–177. ICST.
- Pla-Castells, M., Martinez-Durá, J. J., Samper-Zapater, J. J., and Cirilo-Gimeno, R. V. (2015). Use of ict in smart cities. a practical case applied to traffic management in the city of valencia. In *2015 SCSP*, pages 1–4. IEEE.
- Selic, B. (2003). The pragmatics of model-driven development. *IEEE software*, 20(5):19–25.
- Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., and David, B. (2015). A literature survey on smart cities. *Science China Information Sciences*, 58(10):1–18.