ARCHITECTURE FOR INDUSTRY 4.0: RELATED ISSUES ABOUT MODELLING AND SIMULATION

MARCOS A. PISCHING[1, 2], FABRÍCIO JUNQUEIRA[1], DIOLINO J. DOS SANTOS FILHO[1], PAULO E. MIYAGI[1]

[1] *Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos,*
Escola Politécnica da Universidade de São Paulo - USP
Av. Prof. Mello Moraes, 2231, 05508-530, Cidade Universitária, São Paulo, SP, Brasil

[2] *Instituto Federal de Santa Catarina - IFSC*
Rua Heitor Villa Lobos, 222, Bairro São Francisco, Lages, SC, Brasil

E-mails: {marcos.pisching, fabri, diolinos, pemiyagi}@usp.br

Abstract— The fourth industrial revolution, also called Industry 4.0, goes in favour of the consolidation of the smart factory by the creation of smart products, procedures and processes, in a way to offer on-demand personalized services that can meet the needs of individual consumers. It is based on high information and communication technology and the application of cyber-physical systems and the introduction of Internet of Things into the factory. To achieve it, one need of the Industry 4.0 is the communication between products and machines to drive the products during the manufacturing processes and enabling that it can receive personalized services. Although, concrete architectures and standardization for Industry 4.0 is a lack. The purpose of this paper is the presentation of a survey considering the issues about methods and approaches in modelling and simulation of architectures for Industry 4.0 in a way that it can guide the building of an architecture. The research takes into account the architecture design of software, CPS, and IoT, the concepts that are closely linked to Industry 4.0. Using a reference approach of modelling and simulation this paper show common issues about the design, including simulation, of architecture models that can be used in future research.

Keywords— Architecture, Cyber-Physical Systems, Industry 4.0, Internet of Things, Modelling and Simulation.

1 Introduction

To meet the requirements and follow the market fluctuations, the industrial sector undergo constant changes on its production processes and value chain that can be simple or more complex. Over the years, the most severe changes has marked the known industrial revolutions. The twenty-first century is witnessing a new era in the industrial sector, attended by cyber-physical systems (CPS), Internet of Things (IoT) and the use of advanced features based on Information and Communication Technologies (ICT). All these concepts are promoting a revolution in the industrial sector, marking the beginning of the 4th Industrial Revolution, also called the Industry 4.0 (Kagermann *et al.*, 2013) (Lasi *et al.*, 2014) (Gorecky *et al.*, 2014). The Industry 4.0 is an emerging concept (Kagermann *et al.*, 2013) that is focusing on consolidation of smart factory (MacDougall, 2014). In the Industry 4.0 the CPS is responsible for the integration of the real world with the virtual world while the IoT enable the communication between things by ICT, including products and machines from the shop floor (Pisching *et al.*, 2015a) (Pisching *et al.*, 2015b) (MacDougall, 2014) (Gorecky *et al.,* 2014). The Industry 4.0 are focused on design and creation of smart products, procedures and processes, in a way to create on-demand personalized services that can meet the needs of individual consumers (Kagermann *et al.*, 2013) (Pisching *et al.*, 2015a).

The Figure 1 shows an overview of Industry 4.0 and its respective collaborative service network. The Industry 4.0 involves different types of businesses, data and heterogeneous systems, where physical systems such as Consumers, Factories, Partners, etc. are mapped to the virtual world, turning them accessible from anywhere and allowing communication between them through collaborative networks over a cloud-based infrastructure in order to meet common needs (Pisching *et al.*, 2015a). In this scenario, one of the biggest challenge is the definition of standards and architectures to enable enterprise communication and integration by the virtual world (Kagermann *et al.*, 2014). It is applied to all enterprise value chain, including the manufacturing processes. In the manufacturing level the Industry 4.0 expect the interactions between machines and raw materials by IoT technology, also called smart manufacturing. Material and products may have a unique identification (such as a Radio Frequency Identification – RFID tag) to drive their directions during the manufacturing processes (Ungurean *et al.*, 2014) and during the product life cycle. The scope of this research is the communication between raw materials and machines along an assembly line.

According to Kagermann *et al.* (2013) the Industry 4.0 yearns for concrete architectures. Although it has been proposed architectures for CPS and IoT, they need to be applied and evaluated in the context of Industry 4.0. The purpose of this paper is the presentation of a survey considering the issues about methods and approaches in modelling and simulation of architectures for Industry 4.0. The research takes into account the architecture design of software, CPS, and IoT. According to Chung (2004) if a system does not yet exist, or if it is in an early stage, a simulation model can give an idea how well the proposed system will perform. To do a simulation model is necessary to follow some method or procedure (Chung, 2004). For this reason, this paper investigates how the scientific community of this area is doing it. Additionally, it explores the opportunities and related issues about modelling and simulation in the field of Industry 4.0.

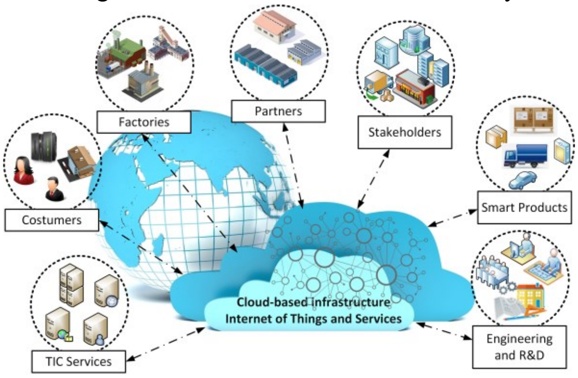

Figure 1 - An overview of Industry 4.0 with collaborative service network (Pisching *et al*., 2015a)

This paper is organized into four sections. The first section presents the contextualization, main concepts, the motivation and paper aim. The section two introduce the review about architecture, including main concepts and architectures for CPS, IoT, additionally it shows issues related to the modelling and simulation. The section three come up with a reflection about modelling and simulation applied in architecture design. Finally, the section four bring forward the conclusions about the survey and directions for future work centred in the architecture for organization and locating services for the Industry 4.0.

2 Literature Review

This section presents a broad view of architecture and an overview about modelling and simulation, centred on software, mainly on CPS and IoT, since there are closely linked to the strategies for Industry 4.0 consolidation. According to Sommerville (2015) an architecture is defined in relation to the components involved in a system and their interconnections.

2.1 Method to Design a Simulation Model

Chung (2004) presents an approach to follow on designing a simulation model. This approach is based on a set of steps that are shown in Figure 2.

The success of project depends on many factors in each stage and should have a very close relationship in the real world, especially in terms of user, business and system.

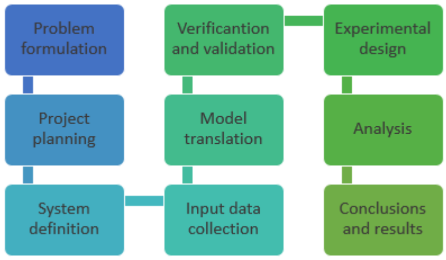

Figure 2 - Steps to design a simulation model (Chung, 2004)

2.2 Software Architecture

In terms of software, architecture is the fundamental structure of the model and the organization of a software system (Sommerville, 2015). According to SEI (2015), a software architecture of a computer system is a representation of the system that contributes in the understanding of its behaviors.

Software architecture is the base for the system and project building, giving the parameters and directions that must be followed by the designers and implementation teams. According to MSDN (2015) software architecture is the process of defining a structure solution that meets all of the technical and operational requirements. It encompasses the set of significant decisions about the organization of a software systems including the selection of the structural elements and their interfaces by which the system is composed. All the systems should be designed based on the integration of three main keys: user, system, and business (Figure 3). During the modelling and design of the software architecture these keys must be considered to provide useful solutions (MSDN, 2015) (Falessi *et al.*, 2011).

According to Falessi *et al.* (2011) software architecture is designed during the early phases of the development process and it facilitates or constrains the achievement of specific functional requirements, non-functional requirements, and business goals.

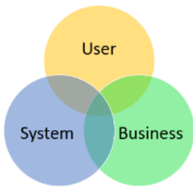

Figure 3 - The main keys for designing software (MSDN, 2015)

To build a software or software architecture generally is used a model of software process that is a set of activities that guide to a product of software (Sommerville, 2015). The Waterfall model, also known as software life cycle, was the first model and it is the basis of many other product life cycles. This method, depicted in Figure 4, is based on 5 steps. Each phase must be concluded before start another, and each stage overlap and exchange information with each other.

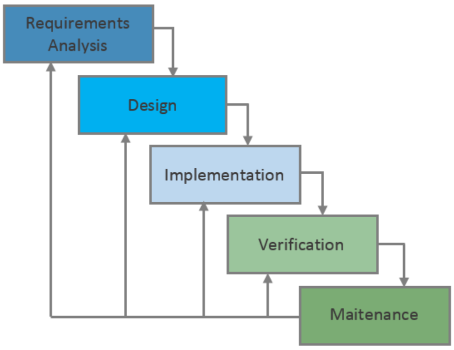

Figure 4 - Waterfall model (adapted from Sommerville, 2015)

2.2.1 Modelling and Simulation for Software Development

Based on Waterfall software development life cycle, Bassil (2012) proposes a simulation model using the Symphony.Net simulation tool in a way to aid project managers to increase the productivity and reducing expenses, workers and hours.

The Bassil's simulation model consist of the following stages: a) divide the Waterfall model into independent phases; b) understand the concept and the requirements that lie behind every phase; c) define the resources tasks, entities, and the work flow of every phase; d) simulate each phase apart and record results; and, e) integrate the whole stages together, simulate the system, and record results. To simulate it the data input was based on a randomly arrivals from a Triangular distribution.

Rus *et al.* (2003) focus their efforts for developing software process simulators and propose a systematic method to the development of discrete simulation-based decision models, and according to the authors this model could be extensible to other modelling approaches. According Rus *et al.* (2003) the life cycle of a simulation model for long-term use is similar to a software, and it is based on three main phases: development, deployment, and operation. So, this stages could be modelled in some life cycle process such as Waterfall. The simulation model of Rus *et al.* (2003) is divided into two categories: engineering activities and management activities. The first activity category comprises a) requirement identification and specification; b) analysis and specification of the modelled process; c) design; d) implementation of the model; and e) verification and validation. The second category is separated into: a) model development planning and tracking; b) measurement of the model and of the model development process; and c) risk management. This simulation model is concerned in all stages of the development software life cycle and it similar to the Chung (2004) approach.

2.3 CPS Architecture

A basic CPS architecture provide components to enable the communication between the real and virtual world (Broy, 2010). The main components and their integration are shown in Figure 5, where "Systems" and "Users" can communicate with the "Mechanical/Physical Systems" respectively by "Connection" interface and "Human-Machine Interface". These interfaces are integrated to an "Embedded System" that can control the machine by the interaction with "Sensors and Actuators" interface that are equipped by appropriate "hardware" and "software" to send and receive data.

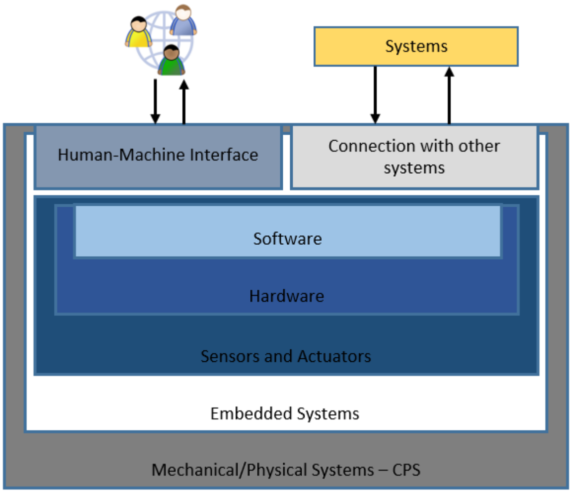

Figure 5 - CPS architecture (adapted from Broy (2010))

Focused on Industry 4.0, Pérez *et al.* (2015) analyse the requirements and propose a model-based architecture and technologies to make the CPS on Industry 4.0 a reality. The authors centre their efforts on a vertical integration and present the requirements and the selection of currently methods, techniques and technologies that are widely used. They consider a vertical integration in a way to collecting data from the production process and convert it into valuable data to remote clients. To capture the data two types of requirements, one related to the user and other related to the implementation point of view. To do it the Model Driven Engineering (MDE) paradigm was used, given the fact that it offer resources for: a) separation of concerns through domain modelling; b) domain model mappings through model-to-model transformation; and, c) automatic generation of configuration files and code through model-to-text transformation. Based on it the authors raised a set of data and technologies to propose the CPS architecture in a way to achieve integration and exchange information. To demonstrate the architecture a case study was presented based on a laboratory application, which the purpose was the illustration of the application to a factory automation.

Lee *et al.* (2015) presents a unified 5-level architecture, named as the 5C architecture, as a guideline for implementation of CPS. This architecture provides a step-by-step guideline for developing and deploying a CPS for manufacturing application focused on Industry 4.0. Their 5C architecture defines how to build a CPS, from the data acquisition and analyses to the final value creation. To do it, the 5C architecture is composed by the following levels: a) smart connection; b) data-to-information conversion; c) cyber; d) cognition; and e) configuration. The 5C architecture guidelines the authors in the data collection from standards and manuals to provide a CPS implementation. The Lee *et al.* (2015) conclusion is that a 5C architecture for CPS in the Industry 4.0 provide a viable and practical guideline for CPS implementation.

2.3.1 Modelling and Simulation of CPS

Jensen *et al.* (2011) present a model-based design methodology considering that it is a powerful technique for CPS. They combine steps such as simulation, software, synthesis, verification and others to propose an explicit and holistic methodology for CPS design from abstraction to architecture, and from concept to realization. The model-based design suggested by Jensen *et al.* (2011) was decomposed into ten steps that is summarized below:

- State the problem – description of the problem to be solved. It must be simple, without the use of mathematics or technical terminology.
- Model physical processes – consists of the basic observations and insight over the relevant physical systems. Models of physical processes are simplified representations of the real world.
- Characterize the problem – determine fixed and adjustable parameters, and variables to be controlled.
- Derive a control algorithm – Determine conditions under which physical processes are controllable and derive a suitable control algorithm to be executed by an embedded computer.
- Select model of computation – The way to define a set of allowable instructions used in a computation along with rules that guide the interaction, communication, and control flow of a set of computational components.
- Specify hardware – hardware definition that must support the environment and provide conditions for interaction with the modelled physical system and implement the control algorithm.
- Simulate – the use of a desktop simulation tool to analyse the behaviour of the problem to be solved.
- Construct - build the device according to specifications.
- Synthesize software – the code implementation. Codes synthesizers may be present in some simulation tools, e.g. LabView.
- Verify, validate, and test - configure adjustable parameters to create test environments. Computational systems may be isolated from physical systems via hardware-in-the-loop testing. Formal verification and validation give insight into the behaviour of an algorithm over all or certain combinations of its inputs, or over the course of time.

To demonstrate the proposed method, the authors apply each stage in a Tunneling Ball Device whose operation demands hardware and real-time embedded computing that deliver high-precision sensing and actuation (Jensen *et al.*, 2011).

2.4 IoT Architecture

A basic IoT architecture comprises at least 4 layers that are shown in Figure 6. The first layer is integrated with hardware to sense and control the physical objects and also for reading data. The second layer provides support to network services and data transfer over the internet. In the third layer entities and services are created and managed in a way to offer services to meet the demands of users and systems. The fourth layer provides methods to the interactions for users and other applications.

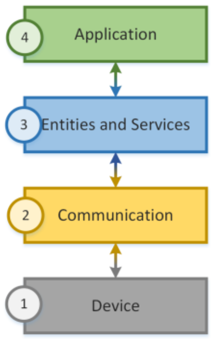

Figure 6 – IoT architecture in four layers (Pisching *et al.*, 2015b) (Xu *et al.*, 2014) (Bandyopadhyay and Sen, 2011)

The IoT concept involves a lot of heterogeneous systems and data that have been generating concerns in the scientific community (Bassi *et al.*, 2013). Worried about it, more than 50 scientists and researchers contributed to the development of IoT-ARM (Internet of Things - Architectural Reference Model), a project funded by the European Union called IoT-A project (Kramp *et al.*, 2013) that was accomplished between 2010 and 2013. The purpose of this project was the creation of a reference model that enable the building of concrete IoT architecture for specific systems. So, according Bassi *et al.* (2013) it is divided into two fronts. First, the creation of a Reference Model for the IoT domain in order to provide a simple understanding. Second, a Reference Architecture that describes essential building blocks as well as design choices to deal with conflicting requirements Although the main project was finished, it is a continuous project that need results for different kind of systems and applications (Bassi *et al.*, 2013).

To develop the IoT-ARM project the authors use the following methodologies:

- Aspect-Oriented Programming – used to define the delineation of functionalities by aspects that is embodied in the concept of functionality group.

- Model-Driven Engineering (MDE) and Model-Driven Architecture (MDA) – adopted to define general concept of transformation from a generic to a more specific model. It was used for describing and developing their best practices.
- Views and perspectives – it was used for the derivation of the IoT Reference Architecture in a way to arrange all aspects of reference architecture according to views and perspectives.

The MDA is a software design approach that was officially launched in 2001 by the Object Management Group (OMG). One of the main aims of the MDA is to separate design from architecture and technologies facilitating that design and architecture can be changed independently. The design addresses the functional (use case) requirements while architecture provides the infrastructure through which non-functional requirements like scalability, reliability and performance are realized. MDA envisages that the platform independent model, which represents a conceptual design realizing the functional requirements, will survive changes in realization technologies and software architectures (SBS, 2015).

By the above methodologies the authors divide the problem in views to provide a reference model and a reference architecture. By the MDE/MDA they build a model independent of technology and platform. The IoT-ARM starting by a contextualization scenario followed by a model definition. The model is organized into model and sub-models with their interactions, which are: a) domain model (definitions and purposes; use cases; model abstraction by UML – Unified Modelling Language; details and related concepts); b) information model (information model definition with relationships, attributes and services); c) functional model (functional decomposition; abstraction levels – functional model and functional vision; definition of functional group; diagram of functional model; diagrams of use case, activities and classes); d) communication model (communication between entities; connection paradigms; interactions – service-to-user, service-to-service, and service-to-resource-to-service; interoperability); and, trust, security and privacy.

The validation of IoT-ARM happens by many use cases but it needs validation by real tests and/or simulation, and is expected that future researches could provide more results of the IoT-ARM application (Bassi *et al.,* 2013).

3 A reflection on model and simulation methods

In this section are presented a comparable view from the related research take into account the Chung (2004) strategy. The comparison is shown in Table 1, where all steps of Chung are listed and compared with other authors.

About the IoT modelling and simulation issues did not have any similar to the Chung (2004) approach. But Bassi *et al.* (2013) use MDA and other techniques to design the Reference Model and Reference Architecture for IoT, which are implicit steps to model and design a system, specially an architecture.

Table 1 - Comparison of strategies to model a simulation project

Chung (2004) approach	**Bassil (2012)**	**Rus *et al.* (2003)**	**Jensen *et al.* (2011)**	**Bassi *et al.* (2013)**
Problem formulation	●	●	●	●
Project planning	●	●	●	●
System definition	●	●	●	●
Input data collection		●	●	●
Model translation	●	●	●	
Verfication and validation	●	●	●	
Experimental design		●		
Analysis		●		
Conclusions and results	●	●	●	●

4 Conclusions

This survey shows some methods to design and model a simulation project. It also can be applied to propose and validate an architecture for Industry 4.0.

The MDA, Bassi *et al.* (2013), and Chung (2004) approach provide the ways to design an architecture to the organization and locating of services for the Industry 4.0 enabling the communication between products (raw materials) and machines. It can also help in building architectures that offer features to drive the products (raw materials) automatically during the manufacturing processes, in a way to provide personalized service and meet the customer individual needs.

Therefore, in the field of Industry 4.0, there are opportunities for modelling and simulation. It is clearly showed that although the researchers were proposing separately methods for software, CPS and IoT, the methods could be used to provide Industry 4.0 solutions, including the communication between products and machines.

From the issues that were surveyed and discussed in this research, is suggested for future work a proposal of a procedure for modelling and simulation applied to a case study, which should be formalized through a formal language, such as the Petri Net that can also be used to simulate and validate the results.

Acknowledgments

The authors would like to thank the financial support of the Brazilian government agency MEC/CAPES, for funding the DINTER USP-IFSC (*Doutorado Interinstitucional entre a Universidade de São Paulo e o Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina*).

Bibliography

Bandyopadhyay, D. and Sen, J. (2011). Internet of Things: Applications and Challenges in Technology and Standardization. Wireless Personal Communications, Volume 58, pp. 49-69.

Bassi, A., Bauer, M., Fiedler, M., Kramp, T., Kranenburg, R. van, Lange, S., Meissner, S. (2013). Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model. Heidelberg New York Dordrecht London: Springer.

Bassil, Y., 2012. A Simulation Model for the Waterfall Software Development Life Cycle. iJET - International Journal of Engineering & Technology, Vol.2, No. 5; pp. 1-7.

Broy, M., (2010). Cyber-Physical Systems: Innovation Durch Software-Intensive Eingebettete Systeme. Berlin: Springer.

Chung, C. A. (2004). Simulation Modeling Handbook: a practical approach. CRC press, Taylor and Francis Group.

Falessi, D., Cantone, G., Kasman, R. and Kruchten, P. (2011). Decision-Making Techniques for Software Architecture Design: A comparative survey. ACM Computing Surveys, Vol.43, No 4; pp. 1-28.

Gorecky, D., Schmitt, M., Loskyll, M. and Zühlke, D., (2014). Human-Machine-Interaction in the Industry 4.0 Era. INDIN 2014, pp. 289-294.

Jensen, J. C., Chang, D. H. and Lee, E. A., (2011). A Model-Based Design Methodology for Cyber-Physical Systems. 7th International Wireless Communications and Mobile Computing Conference (IWCMC), 2011, pp. 1666-1671.

Kagermann, H., Riemensperger, F., Hoke, D., Helbig, J., Stocksmeier, D., Wahlster, W., Scheer, A.-W. and Schweer, D. (2014). Smart Service Welt. Frankfurt: ACATECH.

Kagermann, H., Wahlster, W. and Helbig, J. (2013). Recommendantions for implementing the strategic initiative Industrie 4.0. Frankfurt: ACATECH.

Kramp, T., Kranenburg, R.V. and Lange, S. (2013). Introduction to the Internet of Things. Em: A. Bassi, et al. eds. Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model. s.l.:Springer, pp. 1-10.

Lasi, H., Kemper, H.-G., Fetkke, P., Feld, T., Hoffmann, M. (2014). Industry 4.0. Business & Information Systems Engineering, Vol.4, pp. 239-242.

Lee, J., Bagheri, B. and Kao, H.-A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manufacturing Letters, Vol. 3, pp. 18-23.

MacDougall, W. (2014). Industrie 4.0 – Smart Manufacturing for the Future. Berlin, Germany.

MSDN, 2015. *MSDN - Microsoft Developer Network.* Available at: https://msdn.microsoft.com/en-us/library/ee658098.aspx, Access in: December, 10th, 2015.

Pérez, F., Irisarri, E., Orive, D. and Marcos, M., (2015). A CPPS Architecture approach for Industry 4.0. 20th Conference on Emerging Technologies & Factory Automation (ETFA), 2015, pp. 1-4.

Pisching, M.A, Junqueira, F., Santos Filho, D.J. and Miyagi, P.E. (2015a). Na Architecture for Organizing and Locating Services to the Industry 4.0. 23rd ABCM International Congress of Mechanical Engineering, COBEM 2015, Rio de Janeiro, pp. 1-8.

Pisching, M.A., Junqueira, F., Santo Filho, D.J. and Miyagi, P.E. (2015b). Service Composition in the Cloud-Based Manufacturing Focused on the Industry 4.0. In: Camarinha-Matos, L.M. *et al.* (2015). Technological Innovation for Cloud-Based Engineering Systems, AICT, IFIP, DOCEIS'15. Caparica, Lisboa, Portugal: Springer, Vol.450, pp-65-72.

Rus, I., Neu, H. and Münch, J., (2003). A Systematic Methodology for Developing Discrete Event Simulation Models of Software Development Processes. Proceedings of the 4th International Workshop on Software Process Simulation and Modeling (ProSim 2003), pp. 1-7.

SBS (2015). What is Model Driven Architecture? (MDA). Select Business Solutions - Software and Services. Available at: http://www.selectbs.com/. Access in December, 22, 2015.

SEI - Software Engineering Institute - Carnegie Mellon University, (2015). Defining Software Architecture. Available at: http://www.sei.cmu.edu/architecture. Access in December, 1st, 2015.

Sommerville, I. (2015). Software Enginnering. 10 ed. Pearson.

Ungurean, I., Gaitan, N.-C., Gaitan, V.G. (2014). An IoT Architecture for Things from Industrial Environment. 10th International Conference on Communication, pp. 1-4.

Xu, L.D., He, W. and Li, S. (2014). Internet of Things in Industries: A survey. IEEE Transactions on Industrial Informatics, Vol.10, N. 4, pp. 2233-2243.