

On Challenges in Engineering IoT Software Systems

Rebeca Campos Motta | Universidade Federal do Rio de Janeiro and LAMIH CNRS UMR 8201 | rmotta@cos.ufrj.br

Káthia Marçal de Oliveira | Université Polytechnique Hauts-de-France LAMIH CNRS UMR 8201 | kathia.oliveira@uphf.fr

Guilherme Horta Travassos | Universidade Federal do Rio de Janeiro | ght@cos.ufrj.br

Abstract

Contemporary software systems, such as the Internet of Things (IoT), Industry 4.0, and Smart Cities represent a technology changing that offer challenges for their construction since they are calling into question our traditional form of developing software. They are a promising paradigm for the integration of devices and communications technologies. It is leading to a shift in the classical monolithic view of development where stakeholders used to receive a software product at the end (that we have been doing for decades), to software systems incrementally materialized through physical objects interconnected by networks and with embedded software to support daily activities. Therefore, we need to revisit the traditional way of developing software and start to consider the particularities required by these new sorts of applications. Since such software systems involve different concerns, this paper presents the results of an investigation towards defining a framework to support the software systems engineering of IoT applications. To support its representation, we evolved the Zachman's Framework as an alternative to the organization of the framework architecture. The filling of such a framework is supported by a) 14 significant concerns of IoT applications, recovered from the technical literature, practitioner's workshops and a Government Report; b) seven structured facets emerged from IoT data analysis, that together represent the engineering challenges to be faced both by researchers and practitioners towards the advancement of IoT in practice.

Keywords: *Internet of things, IoT, Contemporary Software Systems Engineering, Empirical Software Engineering*

1 Introduction

The Internet of Things (IoT) contributes to a new technological revolution affecting society. IoT is a paradigm that allows composing systems from uniquely addressable objects (*things*) equipped with identifying, sensing, or acting behaviors and processing capabilities that can communicate and cooperate to reach a goal. From primary devices with simple software solutions to the large-scale, high-performance software systems producing and analyzing massive amounts of data, IoT is going to reach all areas of interest (Jacobson et al. 2017). Due to its far-reaching potential, IoT can use all kinds of technologies available today and will drive the development of new software systems to solve new problems, some still unknown (Atzori et al. 2010; Jacobson et al. 2017).

Software Engineering, as a discipline, has gone through constant changes since its conception. Several concepts, methods, tools, and standards have been proposed to support the development of software (IEEE 2004; Trappey et al. 2017), and the Internet has given a significant shift in the area. It makes more explicit the need to evolve the software technologies previously proposed to support the building of systems fitting new features.

Systems engineering is a research area embracing multidisciplinary, integrating different disciplines to reach successful systems according to their purposes, including software, which is essential for IoT materialization. Therefore, IoT leads to an era where, rather than developing software, practitioners are going to engineer systems embedding much software into the system's parts. In this scenery, the initial problem of our research is to identify the concerns regarding the development of IoT software systems, and whether the existing software technologies

within the areas (facets) related to engineering such systems are enough for supporting their development. Overall, this paper describes the results of investigations dealing with the road ahead on IoT development. The concerns captured through observations in the technical literature, from practitioners in specific workshops and a national initiative regarding IoT in Brazil pave this road.

The filling of IoT facets combined with the concerns is what we call engineering challenges, capturing knowledge necessary to support a specific activity. The conceptual framework aims to contemplate all facets involved in IoT and present the recovered concerns, by simplifying and organizing their presentation. The Zachman's proposition for information systems architecture (Zachman 1987) (Subsection 2.2) was borrowed and tailored to compose such a framework.

Our motivation to investigate and contribute to the IoT paradigm is therefore supported by its relevance (CNI 2016; Lu 2017) and the need for a holistic approach and multidisciplinary view for the development of new software solutions (Bauer and Dey 2016; Aniculaesei et al. 2018). This reflects in a demand for technical competencies and skills detained by different practitioners to engineer such software systems (Desolda et al. 2017; de Farias et al. 2017) and the lack of specific software engineering methodologies to support IoT (Zambonelli 2016; Larrucea et al. 2017; Jacobson et al. 2017). Some of the challenges are focused on interaction issues, whether it is between humans or things, which is essential for the complete establishment of the paradigm (Motta et al.). In our proposal, we introduce a multifaceted conceptual framework as a step towards addressing some of these issues.

This paper extends (Motta et al. 2018), including more details of the studies, deepen the discussions and providing a usage example of the proposed framework. The paper is

organized as follows: Section 2 presents the context of this research and the Zachman’s framework. Section 3 presents the research strategy followed by the primary results: the IoT concerns (section 4), IoT facets (section 5) and the definition of the framework (section 6) with an example of use. Section 7 concludes the paper with final remarks regarding threats to validity and ongoing works.

2 Conceptual Background

This section starts by presenting the source of motivation for this research, the CACTUS Project. Next, it presents some basic concepts related to the Zachman’s Framework, which is the ground for the proposed Conceptual Framework organizing the results presented in this paper.

2.1 The CACTUS Project

The CNPq CACTUS research project was performed based on the aim of understanding test strategies for quality assessment of actor-computer interaction in context-aware systems, as one of the chief characteristics of ubiquitous systems (Spínola and Travassos 2012; Santos et al. 2017; Matalonga et al. 2017). Research teams from two Brazilian universities (Federal University of Rio de Janeiro and Federal University of Ceará) and one French university (Université Polytechnique Hauts-de-France) worked together in the project.

It started with the assumption that interaction is not limited to human and computers in ubiquitous systems. It encompasses the interaction among different devices, such as sensors, actuators, as well as other systems, which we consider the term actor-computer interaction as more adequate. From the results achieved in the project and the technological evolution of the area, we look IoT as related to Ubiquitous Systems, sharing some characteristics and challenges (Andrade et al. 2017). This work is one of the results of this change of perspective.

2.2 Zachman’s Framework

The Zachman’s Framework (Zachman 1987) was introduced in 1987 to comprehend the scope of control within an enterprise and to provide a holistic view of the enterprise architecture that may be used as a base for its management. It still is an essential reference for enterprise architecture and supported by many types of modeling tools and languages (Goethals et al. 2006).

Zachman’s motivation to develop the framework was that “with increasing size and complexity of the implementation of information systems, it is necessary to use some logical construct for defining and controlling the interfaces and the integration of all of the components of the system” (Zachman 1987). The framework is suitable for working with complex systems, and despite its original purpose, its use is not limited to enterprise architecture. Alongside with that, it has been used to assess the development process (de Villiers 2001), for requirement engineering (de Villiers 2001; Technology 2015), business process modeling (Sousa et al. 2007), to instantiate an IEC standard (Panetto et al. 2007), and applied to Systems of Systems (Bondar et al. 2017). Also, Zhang *et al.* used this framework for safety analysis in Avionics Systems (Zhang et al. 2014). More framework evidence use can be observed in different case studies (Panetto et al. 2007; Nogueira et al. 2013; Aginsa et al. 2016), the latter claiming that “Zachman’s framework continues to represent a modeling tool of great utility and value since it can integrate and align the IT infrastructure and business goals.”

From the data recovered in our research, we realize that concepts and properties related to IoT change according to the context and actors involved. This multifaceted view of IoT shows once again that it is a multidisciplinary paradigm. For this reason, a representation of the concepts should be as comprehensive as possible to represent all aspects involved. This framework is primarily defined considering a table, crossing perspectives, and interrogative questions as presented in Table 1 (Zachman 1987; Sowa and Zachman 1992).

Table 1. Zachman Framework with cells filled showing examples of description (Sowa and Zachman 1992).

		INTERROGATIVE QUESTIONS					
		What	How	Where	When	Who	Why
P E R S P E C T I V E S	Planner	Things important to the business	Process performed	Business locations of operations	Events and Cycles important to the business	Organizations and Agents important to the business	Business goals and strategies
	Owner	Semantic Model	Business Process Model	Business Logistic System	Master Schedule	Workflow Model	Business Plan
	Designer	Logic Data Model	Application Architecture	Distributed System Architecture	Process Structure	Human Interface Architecture	Knowledge Architecture
	Builder	Physical Data Model	System Design	Technology Architecture	Control Structure	Presentation Architecture	Knowledge Design
	Implementer	Data Definition	Program	Network Architecture	Timing Definition	Security Architecture	Knowledge Definition
	User	Data	Function	Network	Schedule	Organization	Strategy

The framework formalization and its conception were presented as a metaphor from the building architecture to system architecture. The perspectives are therefore described as (Sowa and Zachman 1992):

- **Planner** - It corresponds to an executive summary for a planner or investor who wants a system scope estimate, what it would cost, and how it would perform.
- **Owner** – It relates to the enterprise business model, which constitutes the business design and shows the business entities and processes, and how they interact.
- **Designer** – It corresponds to the system model designed by a systems analyst who must determine the data elements and functions representing business entities and processes.
- **Builder** – It refers to the technology model, which must tailor the information system model to the details of the programming languages, I/O devices, or other technologies.
- **Implementer** – It relates to the detailed specifications that are given to programmers who code individual modules without being concerned with the overall context or system structure.
- **User** - The user perspective was added in a later version and represents the view of the functioning building, or system, in its operational environment.

The framework presents six fundamental questions in the columns to outline each perspective and support the answering to the questions regarding:

- **What:** some entity (that can be real-world objects, logical or physical data types).
- **How:** some process.
- **Where:** some location.
- **Who:** some role played by a person or a computational agent.
- **When:** time, a subtype such as a date, or time that is coincident with some event.
- **Why:** some goal or subgoal that provides the reason that motivates the model for that row.

Considering the extensive use of the Zachman Framework for representing different domains and technologies and its flexibility of being customized to represent the complexity in each context, we decided to take it as a basis of our work. To that end, we analyzed concerns and facets related to the multidisciplinary of IoT applications to be used as inputs of information and requirements for its first organization.

2.3 Related Work

In this work, we propose a holistic engineering view, based on the principles of Systems Engineering. In the search, we came across the work of Patel and Cassou that propose a development methodology and framework to support the implementation of IoT applications. Their approach is designed to address essential challenges (lack of division of roles, heterogeneity, scale, different lifecycle phases) that differentiate IoT applications from others (Patel and Cassou

2015). The proposal of their methodology is based on the separation of concerns: domain, functional, deployment, and platform. Each concern has specific steps to guide the development, implemented in a defined process. There are some similarities to our proposal. We highlight their strategy to attack multidisciplinary by using four concerns with a diverse set of skills performed by five different roles. However, our proposal differentiates from that because it offers a broader view of the concerns and focuses more on supporting the development team to move out of the problem domain with an action plan stepping into the solution domain.

Two other works (Alegre et al. 2016) and (Sánchez Guinea et al. 2016) are literature reviews, focusing on engineering strategies to develop Context-Aware Software Systems (CASS) and Ubiquitous Systems, respectively. In (Alegre et al. 2016), the results are based on a literature review, and a survey carried out with specialists in CASS. It presents an extensive work in the CASS area, analyzing and characterizing the concept of context as well as their interaction types and main features. The most exciting part for the perspective of our work is that they search the literature for developing techniques and methods that have been adapted from conventional systems to CASS throughout the most common stages of a development process: Requirements Elicitation, Analysis & Design, Implementation, and Deployment & Maintenance.

None of the techniques presented fully meets the CASS requirements, and the authors conclude the work by recommending a more holistic and unified approach for the development of CASS, arguing that it should be different from the conventional software engineering approach for creating these systems (Alegre et al. 2016).

Another work is from Costa *et al.* (Costa et al. 2017). It presents more than just the requirements and needs of an IoT application, focusing on its challenges and proposing an approach to support the requirements specification of IoT systems named IoT Requirements Modeling Language (IoT-RML). We share some of the motivations with this work since it states that different perspectives and the heterogeneous nature of IoT should be considered in the development of such software systems. The Domain Model composes their proposal for the abstraction and a SysML profile for the specification. In their model, a stakeholder expresses a requirement as a proposition, and the requirement may influence or conflict with other requirements. Their approach supports both functional and nonfunctional requirements, which is crucial in this scenario. Through their solution, four requirements specification activities are supported: elicitation of system's requirements from the stakeholders that will generate an initial model in their tool, the analysis to identify influences and conflicts among requirements updating the model representing them, resolution of conflicts, and the last activity, to decide on a candidate solution containing the requirements to be addressed. A proof of concept is presented to illustrate the approach used in the context of a smart building, focusing on employees' safety and energy efficiency.

Our proposal can somehow be related to the IoT-RML approach (Costa et al. 2017). However, we aim to address the problem understanding in the conceptual phase, which focuses on a step before the specification requirements considering a multi-perspective and multidisciplinary strategy.

Another related work is from Aniculaesei *et al.*, which argues that conventional engineering methods are not adequate for providing guarantees to some of the challenges specific of autonomous systems, such as dependability, the focus of their work (Aniculaesei et al. 2018). Some of the main points discussed is the possibility of adaptive behavior present in IoT, as they adapt their behavior to better interact with other systems and people or to solve problems more effectively, and variations in the context, the formerly closed and valid development artifacts may not capture the changes and be inadequate since the environment and the system behavior can no longer be fully predicted or described in advance (Aniculaesei et al. 2018). In response to these challenges and gaps, the authors propose an approach based on the notion of Dependability Cages. Their approach deals with external risks (uncertainties in the environment) and internal risks (system changing behavior), both at the development and the operation.

At the moment of preparing this manuscript, we observed a lack of more concrete proposals for the materialization of the IoT paradigm. We aim to address the challenges presented in (Alegre et al. 2016) and (Sánchez Guinea et al. 2016), filling the gaps from (Patel and Cassou 2015), (Costa et al. 2017) and (Aniculaesei et al. 2018) focusing on the issue of multidisciplinary and providing support to decision-making in the initial development phase of problem understanding.

3 Research Strategy

Figure 1 presents our investigation strategy. It is composed of three parts and involves performing different lines of investigation and studies. The first part of our investigation regards IoT Concerns. It aims at presenting concerns, issues, and difficulties frequently reported regarding the development of IoT applications. To recover such concerns, we collected data from different sources of information considering a literature review (Subsection 4.1), discussion with practitioners (Subsection 4.2) and reading a Brazilian Government report (Subsection 4.3). Based on the identified concerns, it was possible to observe research gaps and main IoT development issues that need effort in their understanding and evolution. These intermediate results can be useful to researchers looking for research opportunities and practitioners planning the construction of IoT applications.

The literature review also supported the identification of 29 IoT definitions. From this set, we conducted a textual analysis, using **coding** procedures, from Grounded Theory (see section 3.1), to assign concepts to a portion of data (Strauss and Corbin 1990). The result was the identification of **IoT Facets** (Section 4) necessary for IoT materialization, in the sense of being the set of parts composing an IoT

software system. We understand facets as “one side of something many-sided” (Oxford Dictionary), “one part of a subject, a situation that has many parts” (Cambridge Dictionary). These facets are the basis for tailoring the 6x6 matrix of the Zachman’s framework (Zachman 1987).

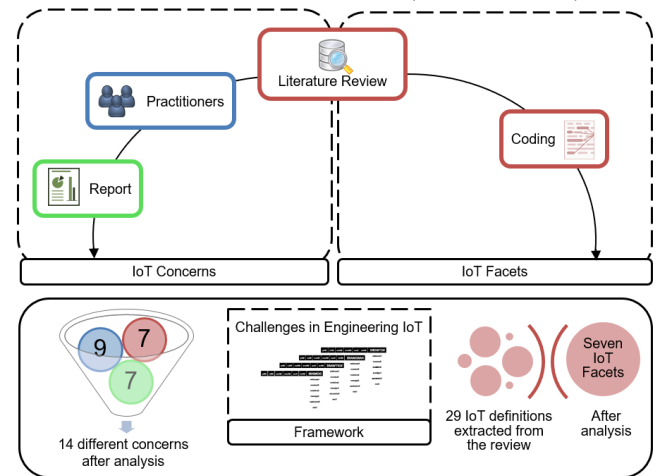


Figure 1: Research Strategy.

The idea of investigating the facets from IoT definitions came in the sense of finding a set of parts composing an IoT software system. It does not try to be exhaustive because due to its far-reaching potential, we do not know to what extent IoT will meet or drive the development of new software technologies to solve new problems. We wanted to differentiate concerns from challenges. Each application alone has a set of concerns that must be addressed with software technologies and other solutions for the software system development or construction. In the case of IoT, we understand that it is a multidisciplinary, multidimensional, and multifaceted paradigm (Gluhak et al. 2011; Gubbi et al. 2013; Jacobson et al. 2017). In this sense, this work presents the IoT facets that must meet the concerns, this being the real engineering challenge (to fill a cell in the framework). The procedures and activities performed for each part are detailed, together with a broader discussion of the results.

However, the **concerns** are somewhat related to the **facets**, and our next activity was to find a way to represent all the concepts that transparently emerged from the sources of information, and that could guide the next research activities. Thus, the last step of our research, as a result of the studies, we introduce a **Conceptual Framework** to organize the challenges of engineering IoT software systems (Section 5).

Our work focusses on discussing the IoT perception and its central issues from three perspectives: technical literature, practitioners, and government, using data collected with different studies. We briefly present the studies and dive into the analyzed challenges, from which we propose a conceptual framework to support the development of such software systems by considering different and complementary facets. This paper presents the research path that led us to the framework, not detailing each study, but instead informing how they inspired a structure composed of six questions, six perspectives and seven facets aiming to define an engineering strategy for IoT development.

3.1 Grounded Theory (GT)

The GT methodology comes as a mechanism to deal with and understand research data and how they relate to each other, considering the IoT domain and features. We rely on these procedures to analyze the recovered data from each study. The principles and procedures of GT according to (STRAUSS; CORBIN, 1998) were used to assist us in developing and analyzing the concepts in this research, as presented below:

- **Planning:** Initially, it identifies the area of interest and the process to be followed inside the GT paths. In our case, each study was planned individually, with the execution and analysis performed by the researchers.
- **Data Collection:** Initially, GT resorted to interviews. However, any method can be used, like focus groups, observations, artifacts, or texts. In our case, we rely on the data extracted from the articles resulting from the literature review, the data recovered from the discussions with practitioners and all the textual documents from the Brazilian government.
- **Coding:** At this step, the researcher should work their sensibility to identify significant data and use the constant comparative analysis method: through iteration, going back and forth in the codes generated observing and comparing to find adequacy, conformity, and coherence among the codes. In our case, the QDA Miner Lite ¹ tool was used to support this part. All the matching from text to code was performed by one researcher and then revised by another. The procedure followed was to review each extraction and the respective code, contributing to the constant comparison until reaching an agreement on the coding.
- **Reporting:** Writing memos, comments, and decision points during the coding phase can enhance the report. Being able to narrate the process of abstraction and describing the rationale behind the codes is the last challenge to sound analysis. In our case, this article comes as the report for the result, from where portions of the extracted data lead to the coding that represents concerns for IoT development.

This approach has been used in Software Engineering research (Seaman 1999; Carver 2007; Badreddin 2013) and was selected since GT provides reference support for the procedures and is adequate to work with a large amount of information. Considering that some concepts have different meanings, this methodology is suitable to establish the similarities and differences among them. The same analysis strategy was used throughout the study.

4 IoT Concerns

Being a multidisciplinary domain, IoT covers many topics from socio-technical to business. We conducted different studies to recover IoT concerns. Each study was planned, considering a specific perspective on the subject. Initially,

we contemplate the academy perspective, recovered through a literature review. Then we decided to broaden the range to represent two other perspectives collected from practitioners and a government report, contributing to a more comprehensive representation. Although they represent different visions, they discuss the same topic. Thus they become complementary, giving us a more comprehensive view of the area.

4.1 Inputs

4.1.1 From the Literature Review

The concerns presented in the technical literature were extracted from a Literature Review (our first empirical study). For the review, we followed the recommendations well established in the literature (Biolchini et al. 2008) focusing on secondary studies since there were already reviews on IoT. The goal of this GQM-based (Basili et al.

1994) review was defined as: *Analyze the Internet of Things domain with the purpose of characterization with respect to definitions, characteristics and application areas from the point of view of software engineering researchers in the context of the available technical literature.*

The selected articles were secondary studies as they rely on primary studies and survey other sources of information to present a bigger picture. presents a research protocol summary. The search engine was Scopus since it indexes several peer-reviewed databases, and is well-balanced regarding coverage and relevance.

Snowballing procedures can mitigate the lack of other search engines and complements the search strategy (Motta et al. 2016; Matalonga et al. 2017). To reduce bias, three researchers executed the review. The process was carried between March and May 2017. Eighty-one articles resulted in the search in Scopus. After the execution of four trials, a selection from title and abstract according to the established criteria, and one level backward/forward snowballing (Wohlin 2014), 12 secondary studies compose the final set. The reviewers read the articles and extracted relevant information according to an extraction form.

We used an extraction form to retrieve the following information from the secondary sources: Reference information, Abstract, IoT definition, IoT related terms, IoT application features, IoT application domain, Development Strategies for IoT, Study Type, Study Properties, Challenges and Article focus. From the discussion of RQ1, we extracted 34 IoT definitions that lead us to understand that IoT is a paradigm allowing the composition of software systems from uniquely addressable objects equipped with identifying, sensing or actuation behaviors and processing capabilities that can communicate and cooperate to reach a goal. Regarding RQ2, we recovered 29 different attributes, where nine of them are discussed with clear evidence from the sources of information. Considering that the results retrieved are from secondary studies, the characteristics

¹ provalisresearch.com/products/qualitative-data-analysis-software/

Table 2. Protocol Summary.

Research questions	(RQ1) What is Internet of Things? (RQ2) Which characteristics define IoT applications? (RQ3) Which are the applications for IoT?	
Search string	Population	("systematic literature review" OR "systematic* review*" OR "mapping study" OR "systematic mapping" OR "structured review" OR "secondary study" OR "literature survey" OR "survey of technologies" OR "driver technologies" OR "review of survey*" OR "technolog* review*" OR "state of research") AND
	Intervention	("internet of things" OR "iot")
Search Strategy	SCOPUS (www.scopus.com) + Snowballing (backward and forward)	
Inclusion Criteria	- To provide an IoT definition; OR to provide IoT properties; OR to provide applications for IoT.	
Exclusion Criteria	- Not provides an IoT definition; AND not provides IoT properties; AND not provides applications for IoT; AND studies in duplicity; AND register of proceedings.	
Study type	Secondary Studies	
Acceptance Criteria	Three distinct readers: - all readers accept => paper is accepted - all readers exclude => paper is excluded - the majority of accept, others in doubt => paper is accepted - else => discuss and consensus	
Technical Report	Detailed information about the planning and execution - https://goo.gl/cZVVDe	

represented reflect more than just the selected set, but rather the whole set of primary studies involved in them, which can strengthen these results. One contribution of the review is to present an organized perspective regarding IoT state-of-the-art. Besides, it allows observing which areas of application are making use of IoT (RQ3).

All of these findings were related and summarized in a report to enrich the IoT paradigm comprehension (see link Table 2). IoT related concepts such as cyber-physical systems (Khaitan and McCalley 2015) and systems of systems (Nielsen et al. 2015) are also discussed in the final report of the review.

The data for discussions and analysis came from part what was extracted from the form, which we treat in this section as concerns. We based our analysis procedure on textual analysis, using codes to assign concepts to a portion of data, identifying patterns from similarities and differences emergent from the extracted data, based on the GT procedures (Strauss and Corbin 1990). It was conducted by two researchers, with crosschecking to achieve a consensus with the analysis to decrease potential misinterpretation and bias. The 12 papers provided 38 excerpts regarding IoT challenges that were organized into seven categories: **Architecture, Data, Interoperability, Management, Network, Security, and Social.**

4.1.2 From Practitioners

Another perspective used to recover IoT concerns was the practitioners' opinion. We performed qualitative studies during two scientific events from which all the participants were developers and/or researchers in the IoT domain. For this reason, we considered them representative, insightful, and experienced in the topic. The following questions guided the discussions in both studies:

- a) Regarding product quality between conventional software and IoT: What is similar? What is different? What needs to be investigated?

- b) Regarding the software technologies between conventional software and IoT: What do we have that can be used directly? What do we have that needs adaptation to be used? What don't we have but need?

The first event (in August 2017) was the 1st *Qualitylot* at the Brazilian Symposium on Software Quality (SBQS). The 21 participants were divided by convenience into groups to deal with the mentioned questions in the following perspectives:

- **People** - Focused on the human end-user. Challenges and impact of this technology in our daily lives, such as social, legal, and ethical. Group of five (5) participants.
- **Product** - Focused on IoT products that can be generated, considering the inclusion of software and "smartness" in general objects and the possibilities of new products in this scenario. Group of nine (9) participants.
- **Process** - Focused on the software development process that should be included in the *things* and consider the big picture of organizing the *things* together. Group of seven (7) participants.

The groups had one hour for discussion. A representative of each group wrote down the main points identified and later presented the ideas for all the workshop participants.

The second event (carried out in September 2017) was a panel in the Brazilian Congress on Software: Theory and Practice (CBSOFT) conducted by the same first event moderator. In this panel, five (5) IoT domain practitioners (experts from academy and industry) and audience were motivated to discuss the same previous study questions. The moderator acted as the reporter in the panel discussion, gathering the central issues, and producing a document reporting the notes.

Next, the notes from both events were collected and analyzed. Besides, open coding procedures based on GT

(Strauss and Corbin 1990) were used and allowed the identification of nine categories of IoT concerns: **Architecture, Interoperability, Professional Quality Properties, Requirements, Scale, Social, Security, and Testing.**

4.1.3 From the Governmental Report

Many initiatives from governments and organizations have demonstrated a growing interest in IoT. In this context, the Brazilian National Bank of Economic Development (BNDES), organized a study to promote economic and social development by analyzing and proposing public policies for IoT. The idea is to obtain an overview of the impact of IoT in Brazil, understanding the country competencies and defining initial aspirations for promoting IoT in Brazil, as it had been documented in a plan of action.

The research is being conducted since the beginning of 2017, and the material² available was used to recover IoT concerns. These results were based on registered initiatives developed by 11 countries and the European Union on IoT, initiatives developed in global scope and interviews with experts to the implementation of the area in Brazil. It was also based on textual analysis on 28 textual documents available. Reading the material allowed extracting information focusing on the presented concerns, analyzing, and similarly organizing them as the two previous information sources (the literature and practitioners). From this, seven categories of IoT concerns emerged: **Data, Interoperability, Network, Professionals, Regulation, Security, and Things.**

4.2 Output: Putting all together

Extracting the perception and concerns regarding IoT from different points of view was essential for the strengthening and direction of our research. For instance, it is possible to observe that, although there are different perspectives, they become complementary to represent the concerns to produce quality software systems. Together, the three sources provided 14 different concerns, which must be met in favor of higher quality IoT software systems (Figure 2).

We present each of the 14 categories with a definition and some example from the input source to support its comprehension:

- **Architecture** - Issues and concerns regarding design decisions, styles, and the structure of IoT systems. Excerpt example: “Finding a scalable, flexible, secure and cost-efficient architecture, able to cope with the complex IoT scenario, is one of the main goals for the IoT adoption.” (Borgia 2014);
- **Data** - It refers to the management of a large amount of data, and how to recover, represent, store, interconnect, search, and organize data generated by IoT from so many different users and devices. Excerpt example: “This new field offers many research challenges, but the main goal of this line of research is to make sense of data in any IoT environment. It has been pointed out that it is always much easier to create data than to analyze them.” (Gil et al. 2016);
- **Interoperability** - Related to the challenge of making different systems, software, and *things* to interact for a purpose. Standards and protocols are also included as issues. Excerpt example: “The end goal is to have Plug n’ Play smart objects which can be deployed in any environment with an interoperable backbone allowing them to blend with other smart objects around them.” (Gubbi et al. 2013);
- **Management** - The application of management activities, such as planning, monitoring, and controlling, in the IoT system will raise the interaction of different *things*. Excerpt example: “IoT is a very complex heterogeneous network, which includes the connections among various types of networks through various communication technologies [...]. Addressing things management is still a challenge.” (Xu et al. 2014);
- **Network** - Technical challenges related to communication technologies, routing, access, and addressing schemes considering the different characteristics of the devices. Excerpt example: “Designing an appropriate topology, routing, and MAC layer is critical for scalability and longevity of the deployed network” (Gubbi et al. 2013);
- **Professionals** - To invest resources in the training of engineers and other professionals can result in the creation of a strategic differential. However, the scenario is different, so more than proficiency in programming languages of lower level; the professional who develops software for IoT should be able to carry out the customization of solutions already developed for specific demands;
- **Quality Properties** - Although some specific properties such as interoperability, privacy, and security are primarily discussed, several other quality attributes are considered different in the IoT domain such as capacity (device and network), installation difficulty, responsiveness, context awareness. Contemplate non-functional requirements by considering what the individual sees, feels and how the things can contribute to that;
- **Regulation** - Governments are working on crucial issues that require significant investment and coordination between the public and private sectors. Within regulatory issues, standardization is one of the most critical, and there is no single strategy to follow. In some cases, it is necessary for the creation of specific laws and institutions regulate privacy and security issues, a topic that is debated today by all the countries mentioned in the report;
- **Requirements** - Considering the IoT nature, with a tendency for more innovation, mainly based on ideas, the requirements can be presented in a less structured

² <https://goo.gl/nmFece>

form. Another concern is that the user can also be a developer since the solutions reach different types of individuals and devices and new features can be attached;

- **Scale** - To develop, manage, and maintain a large-scale software system is a concern. As the number of devices in the software system increases along with the number of relationships, new technologies are needed to maintain a software system with the quality level required.
- **Security** - Issues related to several aspects to ensure data security in IoT systems. For that, a series of properties, such as confidentiality, integrity, authentication, authorization, non-repudiation, availability, and privacy, should be investigated. Excerpt example: “*Security issues are central in IoT as they may occur at various levels, investing technology as well as ethical and privacy issues [...] This is extremely challenging due to the IoT characteristics.*” (Borgia 2014);
- **Social** - Concerns related to human end-user to understand the situation of its users and their appliances. Excerpt example: “*For a lay person to fully benefit from the IoT revolution, attractive and easy to understand visualization have to be created.*” (Gubbi et al. 2013);
- **Testing** - IoT will provide unprecedented universal access to connected devices. Testbed and acceptance tests are sophisticated, and there is a greater need for other types of tests, for example, usability, integrity, security, performance;
- **Things** - For the devices, which includes their access and gateways, there are several non-functional restrictions inherent to IoT that should be present in the products. These restrictions increase the total cost of the objects, such as an energy consumption alternative when it is not possible to connect to the power grid.

It is interesting to notice that the concerns are usually interrelated, confirming the multidisciplinary nature of IoT. For example: “*For technology to disappear from the consciousness of the user, the Internet of Things demands software architectures and pervasive communication networks to process and convey the contextual information to where it is relevant*” (Gubbi et al. 2013), this excerpt is coded for an architectural issue and network as well.

Another example is “*Central issues are making full interoperability of interconnected devices possible, providing them with an always higher degree of smartness by enabling their adaptation and autonomous behavior, while guaranteeing trust, privacy, and security.*” (Atzori et al. 2010), which was coded both for interoperability and for security issues. Provide solutions to the issues presented here can be tricky to achieve due to the diversity of concerns, a variety of devices

We can see that each source has its particularities, and some are consistent with its origin. It is expected that practitioners have a more technical and in-depth view presenting more individual and software-oriented issues regarding IoT software systems. The concerns with **management** and **quality** are transversal to the implementation of such software systems and can be observed in any point of view, but the practitioners have specific concerns of quality, such as meeting non-functional requirements, which bring more specificity and definition to this issue. Also, **requirements** and **testing** issues are still somewhat open on how to represent, describe, and integrate software systems. These three aspects must be met in the software systems regardless of their scale, which in IoT software systems can reach the ultra-large-scale, bringing their associated problems. These three concerns are affected by one aspect that we observed in the literature review. From the characteristics extracted, we could observe that IoT properties and its characterization are not explicit, neither the characteristics that can affect the development process of such applications. Unclear characteristics can impair requirements, which in turn affects the testing, hindering the overall system quality. We consider that this difficulty is partially due to conceptual aspects, since IoT and the related concepts are not yet established and not enclosed by a single definition, being the concept still under discussion (Shang et al. 2016).

Considering the increasing number of interconnected devices, the size or **scale** of IoT software systems can grow consistently. The systems can achieve a more wide-scale coupled with complicated structure-controlling techniques, which brings new challenges to their design and deployment (Huang et al. 2017). New solutions for architectural foundations, orchestration, and management are essential for dealing with scale issues, especially for Ultra Large Scale Systems such as Smart Cities and autonomous vehicles (Roca et al. 2018).

Concerning **regulation**, some actions are being made, from governments³ and other institutions⁴, to form an adequate legal framework. It is necessary to prompt action to provide guidance and decisions regarding governance and how to operate IoT applications in a lawful, ethical, socially and politically acceptable way, respecting the right to privacy and ensuring the protection of personal data (Caron et al. 2016; Almeida et al. 2018).

For the devices, sensors, actuators, tags, smart objects and all the **things** in the Internet of Things, or of Everything, these are some of the aspects that should be taken into consideration: a) resources and energy consumption, since intelligent devices should be designed to minimize required resources as well as costs; b) Deployment since they can be deployed one-time, or incrementally, or randomly depending on the requirements of applications; c) Heterogeneity and Communication: different things interacting with others, they must be available, able to communicate and accessible (Madakam et al. 2015; Li et al. 2015).

³ <https://aioti.eu/> and https://ec.europa.eu/commission/priorities/digital-single-market_en

⁴ <https://www.kiot.or.kr/main/index.nx> and <https://www.digicatapult.org.uk/>

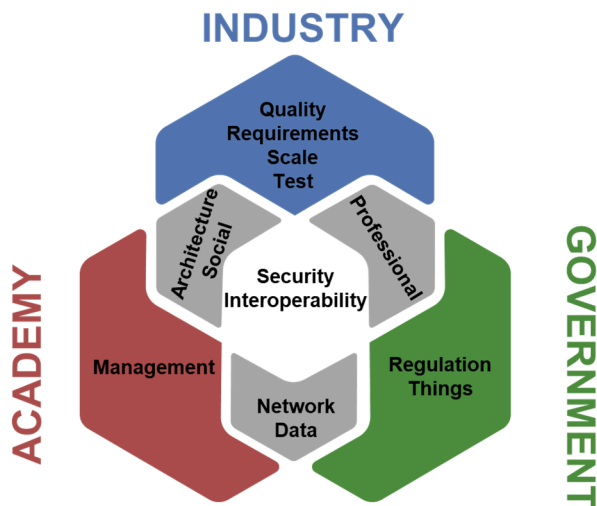


Figure 2. IoT Concerns.

At the intersection between Industry and Literature, we have **architectural** and **social** issues. Both concerns are open due to the area novelty in which there is still an uncovering of how to deal and what to expect. **Architecture** is a recurrent issue in the literature being pointed out by (Liao et al. 2017) as one of the priority areas for action and reported by (Trappey et al. 2017) to be one of the official objectives of ISO/IEC JTC1. In general, the status is that there still no consolidated standard nor well-established terminologies to uniform advancements for architecture in IoT.

Regarding **social** concerns, given that the objects, devices and a myriad of *things* are likely to be connected to many others, being people one of the actors as well (Matalonga et al. 2017), it is necessary to explore the potential sociotechnical impacts of these technologies (Whitmore et al. 2015). Using such devices to provide information *about* and *for* people are one of the applications. Many challenges and concerns should be addressed to achieve the benefits aimed with IoT. In facilitating the development is required the design of data dissemination protocols, and evolve the solutions for privacy, security, trust maintenance, and effective economic models (Guo et al. 2012). As affirmed by (Dutton 2014), if not designed, implemented, and governed in an appropriate way, these new IoT could undermine such core values as equality and individual choice.

At the intersection between Industry and Government, we have the concern of **professionals**, which is represented by the preparation of their skills and knowledge as for the teams that should be multidisciplinary to meet IoT premises. If requirements, testing, and other technical activities are under discussion, we need to think about the professional who will satisfy and perform such activities (Yan Yu et al. 2010). With the development of IoT, different people, systems, and parties will have a variety of requirements; one of the abilities required is how to translate these requirements into new technologies and products. Other skills are related to manage the frequency of information generated, manage the ubiquity and actors involved in interactions, develop and maintain privacy and security

policies (Tian et al. 2018). As the area is new and it is defining the professionals and teams that will work on it too, so it is essential to discuss the professional, develop skills and knowledge necessary for this new generation of innovators, decision-makers and engineers (Kusmin et al. 2017).

Connectivity, Communication, **Network**, and the multiple related concepts that enable the evolution of interconnected objects is a critical point for the materialization of IoT (Gubbi et al. 2013). One of the main challenges of this scenario is a vast amount of information identified, sensed and act upon that must be processed mostly in real- or near-real time with an unobtrusive delivery of personalized manner, ensuring data availability and reliability, the channel between devices, and between the human and devices (Mihovska and Sarkar 2018). Many open challenges require new approaches to a quality network in this scenario. Therefore research should progress into practice to ensure the benefits for the users. Together with Network concerns, we have **Data** issues. In a world with “anytime, anyplace connectivity for anyone and connectivity for anything” (Conti 2006), we can see how quickly data can be generated and how vast amounts of information are created. Some of the challenges are related to the continuous and unstructured creation of connection points (devices, things); the persistence of data objects, unknown scale, and data quality (Uncertainty, Redundancy, Ambiguity, Inconsistency, Incompleteness) (Gil et al. 2016).

However, above these, **security** and **interoperability** concerns are at the center of all IoT related discussions. For IoT, for example, it enables computing capabilities in *things* around us and interoperability is the attribute that enables the interaction among heterogeneous devices, with varied requirements of different applications. Interoperability can range from different levels like technical, syntactical, semantic, and organizational, which varies according to the software system needs. Complete interoperability is an open question for current software and essential for IoT due to its comprehensive nature. Issues like encryption, trust, privacy, and any security-related concerns are of utmost importance since IoT are inserted in someone’s personal life or into the industry. High coverage procedures should guarantee software system security and trustworthiness.

5 IoT Facets

IoT leads to an era where, rather than developing software, we need to engineer software systems embracing multidisciplinary, integrating different areas for the realization of successful products according to their purposes. It means that Software is one of the IoT facets, which, together with others, are necessary for IoT materialization. Aiming at identifying those different facets that characterize this multidisciplinary, we performed an analysis of the IoT definitions identified in the literature review (section 3.1). This analysis was based on GT procedures (Strauss and Corbin, 1990). The 29 extracted IoT definitions were organized in a table with one field of “code” to assign an area, topic, discipline (named here as a facet)

related to a definition excerpt. This coding process was executed by three researchers separately, using separate and independent documents. An example of the document is presented in Figure 3. It is composed of three columns: a) Index: with the definition number; b) Definition: where each definition is presented as extracted from the paper; c) Code: with the codes associated with portions of the definition, with a color scheme to help their identification.

1.	Definition	Code
2.	The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc. – which, through unique identifying schemes, are able to interact with each other and cooperate with their neighbors to reach common goals.	Network/HW Distribution
3.	In fact, "Internet of Things" semantically means "a world-wide network of interconnected objects – usually, identifiable, based on standard communication protocols".	Network/HW Distribution
4.	The definitions above paved the way to the ITU vision of the IoT, according to which: "from anytime, anywhere, connectivity for anyone, we will now have <i>connectivity for anything</i> ".	SW, User/HCI, HW Network
5.	A similar vision is available from documents and communications of the European Commission, in which the most recurrent definition of IoT involves <i>Things having identities and virtual personalities operating</i> in smart spaces using <i>intelligent interfaces</i> to connect and communicate <i>within social, environmental, and user contexts</i> .	SW, User/HCI, HW Network Data
6.	Its members focus on "a world where <i>things can automatically communicate to computers and each other</i> providing services to the benefit of the human kind".	SW, User/HCI, HW Network
7.	IoT refers to an emerging paradigm consisting of a continuum of uniquely addressable <i>things communicating</i> one another to form a worldwide <i>dynamic network</i> . The origin of IoT has been attributed to members of the Auto-ID Center at MIT, the development community of the <i>Radio-Frequency Identification (RFID)</i> , around 2000. Their idea was visionary: they aimed at discovering information about a tagged object by browsing an <i>Internet address or a database entry corresponding to a particular RFID</i> . To address the above idea, they worked on the development of the <i>Electronic Product Code (EPC)</i> , i.e., a universal identifier that provides a unique identity for every physical object, with the aim of spreading the use of RFID in worldwide networks.	SW, User/HCI, HW Network Distribution
8.	Today, the concept of thing is more general and is not limited to RFID only. A thing can be any <i>physical object (tag, RFID sensor, scanner, camera, smart items)</i> but also a virtual/digital entity, which moves in time and space and can be uniquely identified by assigned identification numbers, <i>names and/or location addresses</i> . Therefore, the thing is easily readable, recognizable, locatable, addressable and/or controllable <i>via Internet</i> . Moreover, this new generation of devices is smart things to the embedded electronics allowing them to sense, compute, communicate, and integrate seamlessly with the surrounding <i>environment</i> .	SW, User/HCI, HW Network Distribution
9.	A few years later, members of the same MIT group used again this concept, defining IoT as: <i>"an intelligent infrastructure linking objects, information and people through the computer networks, and where the RFID technology found the basis for its realization"</i> .	SW, User/HCI, HW Network
10.	Specifically, they distinguish three categories: (i) <i>Things oriented</i> , where the focus is on the "objects" and on finding a paradigm able to identify and integrate them; (ii) <i>Internet oriented</i> , where the emphasis is on the networking paradigm and on exploiting the IP protocol to establish an efficient connection between devices, while simplifying IP so that it can be used on devices with very limited capacity; and (iii) <i>Semantic oriented, which aims to use semantic technologies, describing objects and managing data, to represent, store, interconnect, and manage the huge amount of information</i> provided by the increasing number of IoT objects. Authors conclude that IoT is the result of the convergence of these different visions.	SW, User/HCI, HW Network Data
11.	IoT is seen as: <i>"a dynamic global network infrastructure with self capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, virtual</i>	Network SW, User/HCI, HW

Figure 3. Example of a document filed with the definitions and marked with coding.

There were two rounds of discussions, first with two, then with all the three researchers. It was done to discuss the similarity and differences in the coding, support the concepts, and reduce bias, until reaching a consensus. From this analysis, we would like to have a set of facets, based on the data we had so far, and be able to sort among the most used to present a minimal set of areas that must be considered when building an IoT software system. After the documents merge, meetings for discussions were held, some of the discussion was regarding the coding granularity level. For example, *network* and *telecommunication* can all be part of a single facet called *connectivity*, aiming to encompass several concepts and keep the same level of abstraction. As a result of this process, we came to the consensus that an IoT software system should consider seven different facets that are defined below including some examples related to the identified challenges and some potentially used technologies:

- **Connectivity** – the Internet is a relevant concept naturally involved in the IoT paradigm. We argue that it is necessary to have available a medium by which *things* can connect to materialize the IoT paradigm. It is essential some form of connection, a network for the development of solutions, and our idea is not to limit Internet-only connectivity but to be able to cover other media.
 - **Related challenges example:** One of the concerns for connectivity is the traffic management and control to deal with the enormous data generated by these devices and guarantee the quality of service (Bera et al. 2017; Li et al. 2018).

- **Related technologies example:** It uses specific solutions according to the application domain and tries to re-use legacy cellular infrastructure and invest in novel communication solutions. It is mostly based on wireless communication technologies that could be divided into Short-Range, Long-Range, and Cellular-based.
- **Things** - In this sense, it means the *things* by themselves in IoT. Tags, sensors, actuators, and all hardware that can replace the computer, expanding the connectivity reach.
 - **Related challenges example:** To deal with **heterogeneity** and **scale** (Rojas et al. 2017), **distribution** –geographically distributed and sometimes, in inaccessible and critical regions (Chen et al. 2018) as well as **mobility** – IoT devices are not static they tend to move between different coverage areas (Bera et al. 2017), are issues related to requirements to be covered in IoT.
 - **Related technologies example:** Many solutions were combined to build devices like sensors, actuators, smartphones, microcontrollers, interactables, cameras, communication and network enablers, and others. Some systems treat *things* giving a virtual representation of these devices enabling remote access, and control of them.
- **Behavior** - The existence of *things* is not new nor their intrinsic capacities. What IoT provides is the chance of enhancements in the *things*, extending their behaviors. In the beginning, the *things* in IoT systems were objects attached to electronic tags, so these systems present the behavior of Identification. Subsequently, sensors and actuators composing the software systems enabled the Sensing and Actuation behaviors, respectively. It can be necessary the use of software solutions, semantic technologies, data analytics, and other areas to enhance the behavior of *things*.
 - **Related challenges example:** Some emerged behavior cannot be attributed to a single system but results from the interplay of some or all systems in the network. Therefore, each system involved must adjust its behavior according to the common goal, which is an open issue (Brings 2017).
 - **Related technologies example:** The first and most common way to treat behavior is in stages, where the more significant behaviors are constituted by, the smaller ones, with this it is possible to reduce the complexity of taking care of the behaviors. Another way to manage behavior is through the use of state machines (Jackson, 2015; Giammarco, 2017).
- **Smartness** - Smartness or Intelligence is related to Behavior but as to managing or organizing it. It is more referring to orchestration associated with *things* and to

what level of intelligence technology can evolve their initial behavior.

- **Related challenges example:** What makes in many cases a system smart is not only the devices that are used and the decision-making process but the whole solution architecture as well (Atabekov et al. 2015), which leads to an architectural challenge in order to achieve smartness.
- **Related technologies example:** It uses actuators, decision-makers, and acting according to the data autonomously collected and treated to perform some activity in the environment. It uses techniques from artificial intelligence, machine learning, neural networking, and fuzzy logic to deal with the data.
- **Problem Domain** - A problem domain is the area of expertise or application that needs to be examined to solve a problem. IoT software systems are developed to reach a goal for a specific purpose. At this point, we are starting from a goal (problem domain) to reach a solution (software system). Focusing on a problem domain is merely looking at only the topics of interest and excluding everything else. It, in general, directs the objective of that solution.
 - **Related challenges example:** IoT applications development is a multi-disciplined process where knowledge from many concerns intersects. This development assumes that the individuals involved in the application development have similar skills. It is in apparent conflict with the varied set of skills required during the overall process involving this engineering (Patel and Cassou 2015)
 - **Related technologies example:** It varies, but the majority deals with software activities related to analysis, design, and activities to understand the problem domain.
- **Interactivity** - It refers to the involvement of actors in the exchange of information with *things* and the degree to which it happens. The actors engaged with IoT applications are not limited to humans. Therefore, beyond the sociotechnical concerns surrounding the human actors, we also have concerns with other actors like animals and the interactions thing-thing. The degree to which it happens works together with the medium through which *things* can connect (connectivity) so that in addition to being connected, they can understand (interoperability).
 - **Related challenges example:** The wide range of heterogeneity issues introduced by among different IoT devices. Standardization, therefore, is a must but is not enough as no single standard can cover everything, as well as some organizations (manufacturers, software companies), would like to follow different

standards or even proprietary protocols (Dalli and Bri 2016).

- **Related technologies example:** To guarantee communication: HTTP, XMPP, TCP, UDP, CoAP, MQTT, and others. To guarantee to understand: JSON, XML, OWL, SSN Ontology, COCI, and others.
- **Environment** - The problem and the solution are embedded in a domain, an environment, or a context. This facet seeks to represent such an environment and how the context information can influence its use.
 - **Related challenges example:** *Things* can be created, adapted, personalized, and rely on contextual data. The integration of *things* with the social and natural environment can contribute to improving this contextual data and is both a challenge and a research opportunity (Davoudpour et al. 2015).
 - **Related technologies example:** In general, the environments are composed of sensors and actuators to sense and change an ambient state. Technologies like IoT, cloud, smart objects, middleware's, Wireless Sensor Networks, Vehicular Ad-hoc Networks, edge computing, artificial intelligence, machine learning, data mining can be employed on these systems.

Section 6.1 presents the facets regarding an example of a specific IoT application. From the data recovered in our research, we realize that concepts and properties related to IoT change according to the context and actors involved. For this reason, the representation should be as comprehensive as possible to represent all aspects involved, motivating the IoT facets proposition.

6 Defining the Framework

As observed in our investigations, the IoT scenario is covered by **concerns** (discussed in Section 4.4) that are seen and treated according to **facets** (detailed in Section), which leads to challenges for its development. In this context, strategic decisions are essential to the development and need to handle all factors involved in IoT without prejudice to the original software life-cycle concerns with deadlines, costs and quality levels of products and processes (Pfleeger and Atlee 1998; Fitzgerald and Stol 2017). In our proposal, we consider both concerns and facets of IoT development.

In the latest technologies, the software is only one of the components since further development is necessary for requirements representation, data infrastructure, network configuration, and others (Tang et al. 2004). Our aim is regarding the conceptual organization of the recovered data that should consider the requirements of different stakeholders and the activities in the different IoT facets. Having such a conceptual structure, we do not aim that it will guide the software system development but rather to organize the concepts more explicitly and support the decision making when engineering IoT software systems. With this goal in mind, we have identified the Zachman's

Framework (Zachman 1987) as the structure that could support the organization of the concepts. For this reason, we tailored the framework, presented in Section 2, as in Figure 4. There are several definitions for IoT, but the concept refers to a paradigm that allows composing software systems from uniquely addressable objects (*things*) equipped with identification, sensing or action behaviors and processing capabilities that can communicate and cooperate to reach a goal. This understanding encompasses the definitions recovered from the literature review and states the composing and characteristic of IoT. The main difference between traditional software systems regards heterogeneity, scale, and the possibilities inherent to the IoT paradigm.

The Zachman's Framework is generic and flexible enough to be used in different scenarios embedding different points of view, so our choice to use it to organize the information we gathered. Because of the meaning of IoT, we will have the following demands to develop an IoT software system:

- **A paradigm that allows composing systems:** IoT is not just the *things* by themselves. It represents a more substantial aggregate consisting of several parts. It implies that there is not a single IoT solution, but a myriad of options that can derive from the *things* and other systems available. It will require some domain and business-specific strategies.
- **From uniquely addressable objects (*things*):** *Things* should be able to be distinguished using unique IDs, a unique identification for every physical object. It concerns the network solutions and hardware technologies required to devise the composing parts of the IoT paradigm.
- **Equipped with identifying, sensing or acting behaviors and processing capabilities:** Once the object is identified, it is possible to enhance it with personalities and other information and enable it to connect, monitor, manage and control *things*. This understanding implies that depending on the "smartness" degree required for a setting, a software solution can be more robust and involve other technical arrangements, such as artificial intelligence.
- **That can communicate and cooperate:** The other part of the paradigm, alongside with the *things*, is the Internet. The Internet (in a broader sense) is the connection channel of the available *things*. Together with this network solution, *things* should be able to communicate, interchange, share, and other issues. For this, a set of characteristics, such as interoperability, should also be present in the *things*.
- **To reach a goal:** This whole scenario is set for a purpose, for a reason, motivated by something. This primary goal is what will guide the development.

This description leads us to tailor the Zachman's Framework in a faceted scheme; each one represents a facet required for an IoT software system (see Figure 4). We argue that a solution for IoT cannot be done without considering all fundamental paradigm aspects, requiring

multidisciplinary technologies and a diverse team to meet them. We consider the IoT facets to address this multidisciplinary. They were extracted from the literature review and cover a set of dimensions needed to be present, in different degrees, in an IoT software system. This initial set can be extended if needed, as we progressed in the research since it limited to the set of sources dealt with in this research.

Alongside with the facets we have: Perspectives and Communication Interrogatives evolved both from the Zachman Framework (Sowa and Zachman, 1992). The perspectives were divided as control (Business, Executive and User), who support the definition of the problem domain, and construction (Architect, Engineer, Technician, and User) parts, that will specialize the facets to solve the problem. We are considering the user perspective as a hybrid because the future vision is that users have active participation in the construction of IoT solutions (Singh and Kapoor, 2017).

The framework considers all the perspectives involved in the planning, conception, building, usage, and maintaining activities of IoT software systems:

- **Executive Perspective** - It focuses on the system scope and management plans, and how it would relate to a particular context.
- **Business Perspective** - It is concerned with the business models, which constitute business design, how they relate, and how the system will be used.
- **Architect Perspective** - This perspective translates the system model designed and determines the logic behind a system considering data elements, process flows, and functions are representing the business entities and processes.
- **Engineer Perspective** - It corresponds to the technology models, which must tailor the model to the programming languages details, devices, or other required supporting technology.
- **Technician Perspective** - The developer follows detailed specifications to build modules, sometimes without being concerned with the overall context or structure of the system.
- **User Perspective** - It concerns the functioning system in use.

From the guidelines provided in the Zachman's framework, we consider the questions as communication interrogatives for our context since the answer to each question in each perspective, and each facet will give us more direct information leading an engineer closer to the solution specification. These are fundamental questions to outline each perspective:

- **What** - Referring to the information required for the understanding and management of a system. It begins at a high level, and as it advances in the perspectives, the data description becomes more detailed;
- **How** - It relates to translating abstract goals into the definition of its operations using software technologies (techniques, technologies, methods, and solutions);

- **Where** - It concerns the activities location; it can be a geographical distribution or something external to the system;
- **Who** - It describes the roles involved with the systems to deal with the facet development, detailing the representation of each one as it advances;
- **When** - It concerns the effects of time over the system, such as the life cycles, describing the transformations and states of the systems;
- **Why** - It concerns to translate the motivation, goals, and strategies going to what is implemented in the facet.

The *perspectives* in the framework should be mapped and updated according to each IoT facet since different stakeholders are concerned with each area. In the questions part, we seek to keep the original questions and adapt the definitions to be clear and cut for the use of IoT. For instance, the “*what*” is the final product to be delivered by each facet, which in turn can be composed of what each perspective delivers. In software, for example, we have the model built by the software architect, the code made by the developer. They are all part of the final product, the software product. The framework structure will be the combination of these concepts filled in as in Figure 4. Each facet aggregates different knowledge areas, such as software and network. With the simple framework structure, we can organize existing knowledge of software technologies, observing gaps for possible research and development opportunities.

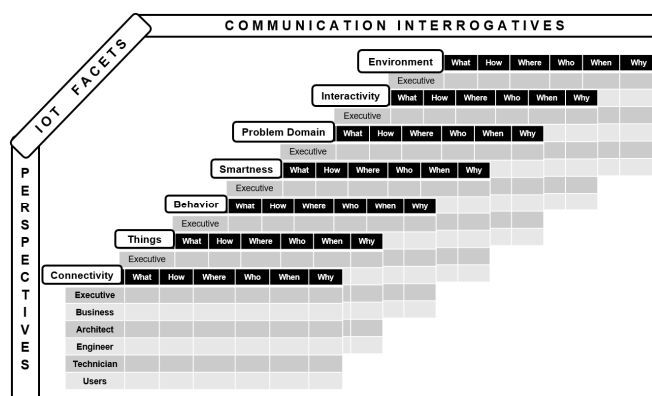


Figure 4. A Framework for engineering IoT applications.

The framework can be filled with knowledge using a bottom-up approach with studies from the technical literature, practitioners, and real cases. We aim to achieve a complete solution on a small scale, to be evolved incrementally. If any adjustment is necessary, we sought to make available the protocols at Delfos⁵ (The Observatory of the Engineering of Contemporary Software Systems) to facilitate access, dissemination, re-execution, and evolution of the findings in order to keep the body of knowledge updated.

The existing knowledge may or may not be enough to cover the IoT paradigm demands, and this must be

investigated from each facet to develop high-quality software systems. Also, each facet will be responsible not only to meet its original premises but to cover the concerns and essential needs of IoT related to that area, such as those presented in this paper. For instance, security and interoperability (the common concerns from the sources) are transversal concerns and must be addressed in the IoT facets related to *things*, behavior, and connectivity. As we evolve the framework structure and deepen our IoT facets of knowledge, we will seek to provide software technologies to meet the concerns as well. The use of the framework can be performed in three steps. By aligning different stakeholders’ perspectives, we want to characterize the problem domain (Step 1). Then, using the framework structure (Figure 4), we aim to extract relevant information for the project (Step 2), that support the definition of decision-making strategy (Step 3). An example of use detailing the steps is presented in the following section.

6.1 Exemplifying the usage of the Framework

Once we have filled the framework with relevant information regarding practices and technologies, considering the different facets and perspectives, we will use it as the basis to support a development strategy. Project information is used to direct and specialize in the framework, to present the concerns that should be taken into account and be used in the strategy to decision making for the specific project.

With this proposal, the goal is not to replace defined activities that are common in the development of traditional software projects. Instead, we hope to address the particularities of IoT projects since they present different and additional characteristics that can bring challenges to its engineering.

This section aims to exemplify the use of the proposed framework. For this, we rely on the results of an IoT software project carried out in the context of an undergraduate software engineering last year discipline of the Computer and Information Engineering course at the Federal University of Rio de Janeiro. Five last periods bachelors’ students with previous knowledge in engineering conventional software systems formed the development team. The course is regularly offered to support the students to work in real problem domain demand and tool-based software engineering environment.

As a case for practice, the development team should provide a software system solution to support the creation of freshwater shrimp in farms. A SEBRAE (Brazilian Service to Support Micro and Small Enterprises)’s claim motivated it: “*Due to the complexity of the production process, and a large number of variables that must be constantly monitored, we suggest the acquisition of software of management, which was not found on the market with enough to be indicated here. Most companies that produce software can provide such a solution, provided that there is a customization of the software.*” A professor (the last author of this paper) and members of the Experimental Software Engineering Group

⁵ <http://146.164.35.157/>

mentored the developers. The software project was executed in the first semester of 2018 and the product (Camarão IoT) deployed in July of 2018. Therefore, the proposal was to idealize and to build an IoT software system to support freshwater carniculture in Brazil.

Based on the described motivation, we present a proof of concept of a solution organized in the structure of the proposed framework. This example enables us to show different facets of arrangements in a basic solution. Because the software system had been implemented and design decisions were taken, we mapped the results to exemplify the use of the framework.

6.1.1 STEP 1 - Define Problem Characterization:

Given the lack of software solutions and the market opportunity for this product, the proposal was to idealize and to build an IoT software system to support freshwater carniculture in Brazil. Our intention with this exemplification was to take what they accomplished and translate it into the proposed framework.

In this characterization step, from the project context, the Executive, Business and User perspectives proposed in the framework are used to support the identification of different concerns and relevant information that must be considered in the solution to be developed. Different roles expressed their expectations regarding the system in the 5WIH structure. The information was condensed and mapped below:

Executive Perspective - The owner represents this perspective and desires a solution that will enable remote and real-time business management. Also, to be able to monitor the overall state of production. Wants to receive notifications of critical conditions and current status, receive periodic reports and estimations, anywhere.

Business Perspective - The manager represents this perspective and wants to receive quick and easy information at any time through the used technologies. Modernize production and have greater control to meet the foreign market. Need to define deadlines and demands, receive information about the water tank, consult stock and production, notifications of critical conditions and current status, receive periodic reports, and estimations in real-time.

User Perspective - Different personas were established for the user perspective, representing the following roles:

- **Installation oversees** - S/He takes care of the installation and stock, reporting back to the manager when it is necessary. S/He needs something that can help the work with clear and direct visual information of when and what actions to take. The system can help to check stock status, receive notification of demands, and notify manager about the need for purchases.
- **Shrimp keeper** - S/He is responsible for preparing the ration and feeding the shrimps. Wants a system that can make the tanks documentation and their characteristics simpler and easier to understand, that would make the job less stressful. Another point that would help in the day-to-day professional life would be to facilitate the

feeding process to avoid repetitive strain injuries. It will be useful to receive feeding schedule, notify biologist shrimp status, visualize tank and shrimp status.

- **Tank keeper** - S/He monitors the tank status, perform measures, and adjust tank conditions. S/He would like to control the tanks more accurately and with a better frequency, without the need to always be running between different tanks. He wants the peace of mind that work is according to the need of the business. Wants to monitor tank status, generate reports, notify critical conditions, secure tank to return to normal conditions, biologist shrimp status, check environmental conditions that can affect the tank and visualize tank and shrimp status.
- **Biologist** - S/He sets the conditions and is responsible for the production health. S/He would like to have past information to be able to perform more precise analyzes and to minimize the error of his estimates, besides being able to compare the evolution of the production in addition to obtaining information about the shrimps in a more accurate and faster way. Wants to update production demand required, update tank conditions to achieve the production demand, define and monitor shrimp health parameters, define and monitor feeding schedule, visualize tank and shrimp status, and generate reports.

As described above, from this step with the framework structure, it is possible to contemplate the different goals for the same solution, thus enriching the initial characterization of the project. Due to the full range of perspectives and goals, the team organizes and prioritizes the primary needs. From this initial part, we defined the primary needs of a system that (1) allows the clear visualization of information regarding the whole process in real-time; (2) support the feeding of shrimp; (3) assist in estimating production and (4) monitor the tank status (Figure 5).

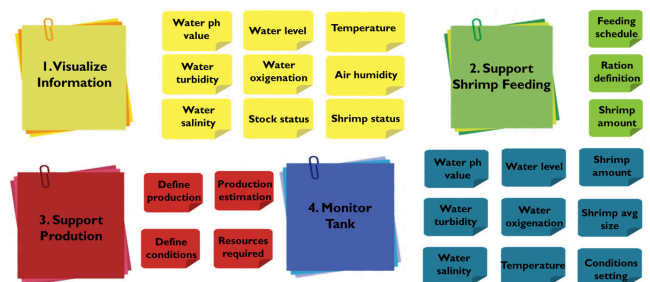


Figure 5. System's needs.

Alongside with the needs presented by the control perspectives, it is required to identify which information can indicate a match with facets, which will support the analysis of the body of knowledge to identify relevant knowledge to engineer a solution to that context.

The Problem Characterization template (used in step 1) will be defined to map the identified system needs with each facet of the body of knowledge in a way that could support the identification of the relevant knowledge. The next

activity of this research in this step is the design of this template. It will comprise the investigation of the concerns defining the facets. A preliminary example of the Problem Characterization artifact for this case study is presented in Figure 6. The idea is to capture the need using questions and perspectives; then we want to map them throughout the artifact, enlightening which concerns should be considered in a given context. It aims to bridge the problem to the facets.

Facet	Concerns	Status
Connectivity	Mobility required	✓
	Short-range communication only	✗
	Limited bandwidth	✗
Things	Heterogeneous devices	✓
	Limited battery and memory resources	✓
	Remote control required	✗
Behavior	Traceability of the device is required	✗
	Capture context information	✓
	Behavior composition	✗
Smartness	Self-adaptation property is required	✗
	Decision-making trigger an action	✓
	Evaluate relevant context information	✓
Interactivity	Semantics influences the interpretation of information	✗
	Cooperation is required	✗
	Seamlessly interaction	✗
Environment	Context influences the solution	✓
	Critical environmental conditions	✗
	Limited infrastructure	✗

Figure 6. A preliminary view of the Problem Characterization for this project.

6.1.2 STEP 2 - Analyzing the Framework

Structure for decision making

After characterizing the problem domain with the characterization artifact, the next step is to analyze the body of knowledge.

In the context of this proposal, the Framework Structure can be seen as a **Body of Knowledge**. This structure was a body of knowledge, has been proposed at a conceptual level, but it has not yet been wholly populated for the IoT systems. Hence, it is one of the next proposed activities, in which we plan to conduct studies to provide evidence-based findings to fill in the Body of Knowledge. Once Body of Knowledge is organized, it can be specialized to the problem context.

For instance, as presented in Figure 5, one of the needs refers to **(4) monitor the tank status**. This feature represents goals from the owner (Executive perspective), manager (Business Perspective), tank keeper, and biologist (User Perspective) and can be developed by different solutions (Table 3). The body of knowledge specialization should assist in the decision-making to implement the desired solution considering this feature’s properties.

Table 3. Possible solutions to monitor tank status.

Solution Option	Description
Manually	The manager defines the required shrimp production and requests production report; he communicates verbally to the biologist. The biologist sets new parameters for the tank and goes to the tank keeper to inform him. The tank keeper manually adjusts tank conditions to meet

	demand. He also manually collects information for the production report and deliver the report to the manager. There is no technical support in the process.
Communication Support	The manager defines the required shrimp production and requests production report; he uses a communication system to inform the biologist. The biologist defines new parameters for the tank and uses the communication system to inform the tank keeper. The tank keeper manually adjusts tank conditions to meet demand. He also manually collects information for the production report and deliver the report to the manager. There is technical support for communication in the process.
Control Support	The manager defines the required shrimp production and requests a production report; he uses a control system. The system notifies the biologist that defines new parameters for the tank. The system notifies the tank keeper that manually adjusts tank conditions to meet demand. He also manually collects information for the production report and make the report available in the system. There is technical support for control in the process.
Sensing Support	The manager defines the required shrimp production and requests a production report using the system. The system notifies the biologist that defines new parameters for the tank. The system notifies the tank keeper that manually adjusts tank conditions to meet demand. He automatically collects information from the sensors for the production report and makes the report available in the system. There is technical support for sensing in the process.
Actuation Support	The manager defines the required shrimp production and requests a production report using the system. The system notifies the biologist that defines new parameters for the tank. The system notifies the tank keeper that uses the system actuators to adjust the tank conditions to meet demand. He automatically collects information from the sensors for the production report and makes the report available in the system. There is technical support for actuation in the process.

The solutions presented are simplified in high-level but are only to exemplify the variety of options dependent on technology, to a greater or lesser degree. For example, if we choose the Sensor Support solution, exemplified in Table 7, it can analyze which relevant knowledge from the body of knowledge should be taken into account, as shown in Table 4. In order to support decision-making to guide the choice and development of the proposed solution, the body of knowledge aims to present the practices and technologies that allow engineers to develop the chosen solution.

Table 4. Some examples of possible practices and technologies from the body of knowledge.

Sensing Support	Connectivity	Bluetooth Low Energy, ZigBee, Z-Wave, NFC (Near Field Communication), RFID (Radio-Frequency Identification), Wi-Fi as enabling technologies, Low-Power Wide-Area technologies, SigFox, Ingenu-RPMA (Random Phase Multiple Access), 2G, 3G, 4G, Software-Defined Network (SDN) and Network Function Virtualization (NFV) and others.
	Things	Temperature sensor TTC104, temperature sensor DS18B20, luminosity sensor LDR 5mm, rain sensor FC37, rain sensor GROVE, humidity and temperature sensor RHT03, Gravity pH Sensor, and others.

	Behavior	Collect water ph-value, water level, water turbidity, water oxygenation, water salinity, water temperature.
--	-----------------	-------------------------------------------------------------------------------------------------------------

6.1.3 STEP 3 – Generate decision-making

Strategy

The output from the previous step (Table 4) should be presented as a set of software practices and technologies, with options of the body of knowledge specialization, and will compose the strategy to support decision-making to drive the solution.

From the problem domain established (the context), the team started the solution engineering. The project was conducted with the team working together. Therefore, there was no formal division of work for construction roles such as Architect, Engineer, and Technician perspectives. They worked as a group to achieve the expected results. For this reason, in this exemplification, we cannot represent different perspectives. It is for illustration purposes and does not allow to demonstrate the full framework potential yet, which is a crucial issue in the continuity of this research.

The solution implemented for the problem (4) monitor the status of the tank is presented in Figure 7 and was implemented in a floater. The floater collects data from the water tank where it was deployed, and it works at each determined interval of time. An operator can adjust the frequency in which the dashboard will update the information received from the sensors, implemented in the floater. A dashboard panel was implemented to enable the visualization of the data collected by the floater and attends the problem stated in (1) allows the clear visualization of information regarding the whole process in real-time. In this context, it is a technological arrangement for data exhibition, where the data producers are the sensors in the floater, which through the connectivity with Wi-Fi can share data with the dashboard to exhibit the data. The overall Floater solution encompasses (Figure 7):

- **Behaviors:** sensing and data collection to collect water level, water turbidity, and water temperature as well as processing, to provide data for the dashboard.
- **Things:** the water level sensor, water turbidity sensor, water temperature sensor, and water salinity were implemented in an Arduino board that worked as the processing unit.
- **Interactivity:** interacts with the dashboard to provide data.
- **Connectivity:** supports the provision of data for the dashboard, implemented by a Wi-fi module in the Arduino.
- **Environment:** the water tank was the environment settled for the sensors to collect data and the network layer used for connectivity.

FLOATER	What	How	Where	Who	When	Why
	Collect water level	Water level sensor	Water tank	Arduino	Schedule time	Monitor tank status
	Collect water turbidity	Turbidity sensor	Water tank	Arduino	Schedule time	Monitor tank status
	Collect water temperature	Temperature sensor	Water tank	Arduino	Schedule time	Monitor tank status
	Collect water salinity	Salinity sensor	Water tank	Arduino	Schedule time	Monitor tank status
	Provide data	Dashboard WIFI	Network Layer	Arduino	Anytime	Enable status visualization

Figure 7. Floater solution implemented for the need (4).

It is necessary to emphasize that the previously described context is only a simplified example of using the framework and does not represent its full use. It was used for illustrating, in a real case scenario, how the different facets overlap and impact each other. It requires a multidisciplinary view of the problem and an adequate development strategy to embrace different disciplines and skills for the accomplishment of successful IoT software systems. We understand that more research is needed to address the open points and to evolve the proposal in general. It represents future tasks to be conducted throughout the continuity of this research.

7 Final Remarks

The emergence of IoT software systems brings new challenges in software engineering. To address these challenges, we should change our way of developing a software system from a monolithic structure to a broader multidisciplinary approach. This paper has presented the results obtained by analyzing data acquired through different strategies, which identified challenges in engineering IoT software systems and the initial results of a conceptual framework to support its development. First, we identified concerns from the technical literature, practitioners, and a government report. Next, we presented the facets that compose IoT software systems, derived from a qualitative study. These results can support practitioners in evaluating risks to construct IoT applications and highlight some research opportunities for researchers. Then we presented a conceptual framework, a way of summarizing the results of the executed studies and structurally present the multidisciplinary of IoT. This structure shall be filled with the existing knowledge of software technologies. Empty cells can identify current technology gaps to engineer IoT software systems. The contribution of this work explains a set of concerns that need to be investigated, showing that it is necessary to distinguish this new software system from traditional ones. Also, the work evolves the Zachman’s framework, to allow the necessary multi-facets representation.

7.1 Threats to validity

The **literature review** used only Scopus as a search engine so that it may be missing some relevant studies. However, from our experience, it can give reasonable coverage when performing together snowballing procedures - backward and forward (Matalonga et al. 2015; Motta et al. 2016). Data extraction and interpretation biases were mitigated with crosschecking between two researchers and by having a third researcher to revise the results. All phases of this review were peer-revised; any doubt was discussed

among the readers, to reduce selection bias. We have not performed a Quality Assessment regarding the research methodology of the selected studies due to the lack of information in the secondary reports. Therefore, it is a threat to this study validity. However, the triangulation with data acquired of practitioners and information extracted from the government report strengthened the representativeness of data and reduced the researchers' bias, powering the results.

From both the data collected from **practitioners** and the **government**, the interpretation of data was supported by the practices of GT, which allowed to get consistency among researchers and shared an understanding of the central concepts. However, other perspectives could be used for data interpretation, imposing a risk of changing the results. It represents a threat to any qualitative study and constitutes a menace that we cannot completely mitigate.

7.2 Ongoing works

We foresee some scenarios of proposed framework utilization. As envisioned contributions of its use, initially, we expect the production of scientific research which considers essential knowledge to practitioners concerning the problem domain. Such knowledge will compose the body of knowledge, which can be useful for both researchers and practitioners sharing and exchanging it. We consider that the evidence-based facets and perspectives have the potential to support the collection of various practices and technologies that can be used in IoT. We expect that the more a facet is filled in a given perspective in response to a question (for example, response to the how, in the engineering perspective, in the behavior facet) more evidence of information will exist about it, which aids the decision-making in practice. In turn, the lack of answers (for example, an empty cell in the body of knowledge) may represent a research opportunity for the academy. In this sense, opportunities and risks are opposites, since an opportunity for researchers is a risk for practitioners.

Once we have filled the cells, it is possible that some will continue empty, since current technologies do not meet IoT needs, representing research opportunities; and both practice and research can get a better observation of what we know and do not know regarding development, since it will allow visualizing where the engineering stands regarding IoT.

Our next steps include the filling of the facets in the manner proposed by Zachman (Sowa and Zachman 1992). However, our primary research aims to fill the cells in the matrix. We conjecture that some of the slots will be empty or partially filled, which means the available software technologies will not support such activity in the way required for IoT. Therefore, it can represent research and development opportunities, which are necessary for the establishment of IoT as a reality. Another conjecture is that some of the concerns can repeat themselves in different slots and different facets, what we call transversal challenges. These cross-sectional slots represent broader concerns that should cover the IoT software system as a whole, for example, security and interoperability issues. We aim to investigate transversal challenges in the nearest future. After

that, we plan to evaluate and refine this conceptual framework.

8 Declarations

Availability of data and materials

Details of the protocol are available in <https://goo.gl/cZVVDC>.

Competing interests

The authors declare that they have no competing interests.

Funding

We thank CNPq for the grant. Professor Travassos is a CNPq Researcher. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Consent for participation and publication

Not applicable.

Acknowledgments

Not applicable.

References

- Agina A, Matheus Edward IY, Shalannanda W (2016) Enhanced information security management system framework design using ISO 27001 and Zachman framework - A study case of XYZ company. In: 2016 2nd International Conference on Wireless and Telematics (ICWT). IEEE, pp 62–66
- Alegre U, Augusto JC, Clark T (2016) Engineering context-aware systems and applications: A survey. *J Syst Softw* 117:55–83. doi: 10.1016/j.jss.2016.02.010
- Almeida VAF, Doneda D, Moreira da Costa E (2018) Humane Smart Cities: The Need for Governance. *IEEE Internet Comput* 22:91–95. doi: 10.1109/MIC.2018.022021671
- Andrade RMC, Carvalho RM, de Araújo IL, et al. (2017) What Changes from Ubiquitous Computing to Internet of Things in Interaction Evaluation? In: International Conference on Distributed, Ambient, and Pervasive Interactions. pp 3–21
- Aniculaesei A, Grieser J, Rausch A, et al. (2018) Towards a holistic software systems engineering approach for dependable autonomous systems. In: Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems - SEFAIS '18. ACM Press, New York, New York, USA, pp 23–30
- Atabekov A, Starosielsky M, Lo DC-T, He JS (2015) Internet of Things-Based Temperature Tracking System. In: 2015 IEEE 39th Annual Computer Software and Applications Conference. IEEE, pp 493–498
- Atzori L, Iera A, Morabito G (2010) The Internet of Things: A survey. *Comput Netw* 54:2787–2805. doi: 10.1016/j.comnet.2010.05.010

- Badreddin O (2013) Thematic Review and Analysis of Grounded Theory Application in Software Engineering. *Adv Softw Eng* 2013:1–9. doi: 10.1155/2013/468021
- Basili VR, Caldeira G, Rombach HD (1994) Goal Question Metric Paradigm
- Bauer C, Dey AK (2016) Considering the context in the design of intelligent systems: Current practices and suggestions for improvement. *J Syst Softw* 112:26–47. doi: 10.1016/j.jss.2015.10.041
- Bera S, Misra S, Vasilakos A V. (2017) Software-Defined Networking for the Internet of Things: A Survey. *IEEE Internet Things J* 4:1994–2008. doi: 10.1109/JIOT.2017.2746186
- Biolchini J, Mian PG, Candida A, Natali C (2008) *Software and Data Technologies*. Springer Berlin Heidelberg, Berlin, Heidelberg
- Bondar S, Hsu JC, Pfouga A, Stjepandić J (2017) Agile digital transformation of System-of-Systems architecture models using the Zachman framework. *J Ind Inf Integr* 7:33–43. doi: 10.1016/j.jii.2017.03.001
- Borgia E (2014) The Internet of Things vision: Key features, application, and open issues. *Comput Commun* 54:1–31. doi: 10.1016/j.comcom.2014.09.008
- Brings J (2017) Verifying Cyber-Physical System Behavior in the Context of Cyber-Physical System-Networks. In: 2017, IEEE 25th International Requirements Engineering Conference (RE). IEEE, pp 556–561
- Caron X, Bosua R, Maynard SB, Ahmad A (2016) The Internet of Things (IoT) and its impact on individual privacy: An Australian perspective. *Comput Law Secure Rev* 32:4–15. doi: 10.1016/j.clsr.2015.12.001
- Carver J (2007) The Use of Grounded Theory in Empirical Software Engineering. In: *Empirical Software Engineering Issues. Critical Assessment and Future Directions*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 42–42
- Chen G, Tang J, Coon JP (2018) Optimal Routing for Multihop Social-Based D2D Communications in the Internet of Things. *IEEE Internet Things J* 5:1880–1889. doi: 10.1109/JIOT.2018.2817024
- CNI CN da I (2016) Indústria 4.0: novo Desafio para an indústria Brasileira. *Indicadores CNI* 17:13
- Conti JP (2006) ITU Internet Reports 2005: The Internet of Things. *Commun Eng* 4:20. doi: 10.1049/ce:20060603
- Costa B, Pires PF, Delicato FC (2017) Specifying Functional Requirements and QoS Parameters for IoT Systems. In: 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech). IEEE, pp 407–414
- Dalli A, Bri S (2016) Acquisition devices in the internet of things: RFID and sensors. *J Theor Appl Inf Technol* 90:194–200
- Davoudpour M, Sadeghian A, Rahnama H (2015) Synthesizing social context for making the Internet of Things environments more immersive. In: 2015 6th International Conference on the Network of the Future (NOF). IEEE, pp 1–5
- de Farias CM, Brito IC, Pirmez L, et al. (2017) COMFIT: A development environment for the Internet of Things. *Future Gener Comput Syst* 75:128–144. doi: 10.1016/j.future.2016.06.031
- de Villiers D (2001) Using the Zachman Framework to Assess the Rational Unified Process Overview of the Zachman Framework. *Ration Edge*
- Desolda G, Ardito C, Matera M (2017) Empowering End Users to Customize their Smart Environments. *ACM Trans Comput-Hum Interact* 24:1–52. doi: 10.1145/3057859
- Dutton WH (2014) Putting things to work: Social and policy challenges for the Internet of things. *Info* 16:1–21. doi: 10.1108/info-09-2013-0047
- Fitzgerald B, Stol K-J (2017) Continuous software engineering: A roadmap and agenda. *J Syst Softw* 123:176–189. doi: 10.1016/j.jss.2015.06.063
- Gil D, Ferrández A, Mora-Mora H, Peral J (2016) Internet of Things: A Review of Surveys Based on Context-Aware Intelligent Services. *Sensors* 16:1069. doi: 10.3390/s16071069
- Gluhak A, Krco S, Nati M, et al. (2011) A survey on facilities for experimental internet of things research. *IEEE Commun Mag* 49:58–67. doi: 10.1109/MCOM.2011.6069710
- Goethals FG, Snoeck M, Lemahieu W, Vandembulcke J (2006) Management and enterprise architecture clic The FAD(E)E framework. *Inf Syst Front* 8:67–79. doi: 10.1007/s10796-006-7971-1
- Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener Comput Syst* 29:1645–1660. doi: 10.1016/j.future.2013.01.010
- Guo B, Yu Z, Zhou X, Zhang D (2012) Opportunistic IoT: Exploring the social side of the internet of things. In: *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, pp 925–929
- Huang J, Duan Q, Xing C-C, Wang H (2017) Topology Control for Building a Large-Scale and Energy-Efficient Internet of Things. *IEEE Wirel Commun* 24:67–73. doi: 10.1109/MWC.2017.1600193WC
- IEEE (2004) *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society Press
- Jacobson I, Spence I, Ng P-W (2017) Is there a single method for the internet of things? *Commun ACM* 60:46–53. doi: 10.1145/3106637
- Khaitan SK, McCalley JD (2015) Design Techniques and Applications of Cyberphysical Systems: A Survey. *IEEE Syst J* 9:350–365. doi: 10.1109/JSYST.2014.2322503
- Kusmin M, Saar M, Laanpere M, Rodriguez-Triana MJ (2017) Work in progress — Smart schoolhouse as a data-driven inquiry learning space for the next generation of engineers. In: 2017 IEEE Global Engineering Education Conference (EDUCON). IEEE, pp 1667–1670

- Larrucea X, Combelles A, Favaro J, Taneja K (2017) Software Engineering for the Internet of Things. *IEEE Softw* 34:24–28. doi: 10.1109/MS.2017.28
- Li S, Xu L Da, Zhao S (2015) The internet of things: a survey. *Inf Syst, Front* 17:243–259. doi: 10.1007/s10796-014-9492-7
- Li S, Xu L Da, Zhao S (2018) 5G Internet of Things: A survey. *J Ind Inf Integr* 10:1–9. doi: 10.1016/j.jii.2018.01.005
- Liao Y, Deschamps F, Loures E de FR, Ramos LFP (2017) Past, present, and future of Industry 4.0 - a systematic literature review and research agenda proposal. *Int J Prod Res* 55:3609–3629. doi: 10.1080/00207543.2017.1308576
- Lu Y (2017) Industry 4.0: A survey on technologies, applications, and open research issues. *J Ind Inf Integr* 6:1–10. doi: 10.1016/j.jii.2017.04.005
- Madakam S, Ramaswamy R, Tripathi S (2015) Internet of Things (IoT): A Literature Review. *J Comput Commun* 03:164–173. doi: 10.4236/jcc.2015.35021
- Matalonga S, Rodrigues F, Travassos G (2015) Challenges in Testing Context-Aware Software Systems. In: 9th Workshop on Systematic and Automated Software Testing 2015. Belo Horizonte, Brazil, pp 51–60
- Matalonga S, Rodrigues F, Travassos GH (2017) Characterizing testing methods for context-aware software systems: Results from a quasi-systematic literature review. *J Syst Softw* 131:1–21. doi: 10.1016/j.jss.2017.05.048
- Mihovska A, Sarkar M (2018) *New Advances in the Internet of Things*. Springer International Publishing, Cham
- Motta RC, de Oliveira KM, Travassos GH (2018) On challenges in engineering IoT software systems. In: *Proceedings of the XXXII Brazilian Symposium on Software Engineering - SBES '18*. ACM Press, New York, New York, USA, pp 42–51
- Motta RC, de Oliveira KM, Travassos GH A Framework to Support the Engineering of Internet of Things Software Systems. *EICS 19 June* 18–21 2019 Valencia Spain 6. doi: 10.1145/3319499.3328239
- Motta RC, Oliveira KM de, Travassos GH (2016) Characterizing Interoperability in Context-Aware Software Systems. In: *2016 VI Brazilian Symposium on Computing Systems Engineering (SBESC)*. IEEE, pp 203–208
- Nielsen CB, Larsen PG, Fitzgerald J, et al. (2015) *Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions*. *ACM Comput Surv* 48:1–41. doi: 10.1145/2794381
- Nogueira JM, Romero D, Espadas J, Molina A (2013) Leveraging the Zachman framework implementation using the action – research methodology – a case study: aligning the enterprise architecture and the business goals. *Enterp Inf Syst* 7:100–132. doi: 10.1080/17517575.2012.678387
- Panetto H, Baïna S, Morel G (2007) Mapping the IEC 62264 models onto the Zachman framework for analyzing products information traceability: A case study. *J Intell Manuf* 18:679–698. doi: 10.1007/s10845-007-0040-x
- Patel P, Cassou D (2015) Enabling high-level application development for the Internet of Things. *J Syst Softw* 103:62–84. doi: 10.1016/j.jss.2015.01.027
- Pfleeger SL, Atlee JM (1998) *Software engineering: theory and practice*. Pearson Education India
- Roca D, Milito R, Nemirovsky M, Valero M (2018) *Fog Computing in the Internet of Things*. Springer International Publishing, Cham
- Rojas RA, Rauch E, Vidoni R, Matt DT (2017) Enabling Connectivity of Cyber-physical Production Systems: A Conceptual Framework. *Procedia Manuf* 11:822–829. doi: 10.1016/j.promfg.2017.07.184
- Sánchez Guinea A, Nain G, Le Traon Y (2016) A systematic review on the engineering of software for ubiquitous systems. *J Syst Softw* 118:251–276. doi: 10.1016/j.jss.2016.05.024
- Santos I de S, Andrade RM de C, Rocha LS, et al. (2017) Test case design for context-aware applications: Are we there yet? *Inf Softw Technol* 88:1–16. doi: 10.1016/j.infsof.2017.03.008
- Seaman CB (1999) Qualitative methods in empirical studies of software engineering. *IEEE Trans Softw Eng* 25:557–572. doi: 10.1109/32.799955
- Shang X, Zhang R, Zhu X, Zhou Q (2016) Design theory, modeling, and the application for the Internet of Things service. *Enterp Inf Syst* 10:249–267. doi: 10.1080/17517575.2015.1075592
- Sousa P, Pereira C, Vendeirinho R, et al. (2007) Applying the Zachman Framework Dimensions to Support Business Process Modeling. In: *Digital Enterprise Technology*. Springer US, Boston, MA, pp 359–366
- Sowa JF, Zachman JA (1992) Extending and formalizing the framework for information systems architecture. *IBM Syst J* 31:590–616. doi: 10.1147/sj.313.0590
- Spínola RO, Travassos GH (2012) Towards a framework to characterize ubiquitous software projects. *Inf Softw Technol* 54:759–785. doi: 10.1016/j.infsof.2012.01.009
- Strauss A, Corbin J (1990) *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications, Inc, Newbury Park
- Tang A, Jun Han, Pin Chen (2004) A Comparative Analysis of Architecture Frameworks. In: *11th Asia-Pacific Software Engineering Conference*. IEEE, pp 640–647
- Technology I (2015) Requirement Formalization using OWL Ontology-based Zachman Framework
- Tian B, Yu S, Chu J, Li W (2018) Analysis of Direction on Product Design in the Era of the Internet of Things. *MATEC Web Conf* 176:01002. doi: 10.1051/mateconf/201817601002
- Trappey AJC, Trappey C V., Hareesh Govindarajan U, et al. (2017) A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. *Adv Eng Inform* 33:208–229. doi: 10.1016/j.aei.2016.11.007
- Whitmore A, Agarwal A, Da Xu L (2015) The Internet of Things—A survey of topics and trends. *Inf Syst, Front* 17:261–274. doi: 10.1007/s10796-014-9489-2

- Wohlin C (2014) Guidelines for snowballing in systematic literature studies and a replication in software engineering. Proc 18th Int Conf Eval Assess Softw Eng - EASE 14 1–10. doi: 10.1145/2601248.2601268
- Xu L Da, He W, Li S (2014) Internet of Things in Industries: A Survey. IEEE Trans Ind Inform 10:2233–2243. doi: 10.1109/TII.2014.2300753
- Yan Yu, Jianhua Wang, Guohui Zhou (2010) The exploration in the education of professionals in applied Internet of Things Engineering. In: 2010 4th International Conference on Distance Learning and Education. IEEE, pp 74–77
- Zachman JA (1987) A framework for information systems architecture. IBM Syst J 26:276–292. doi: 10.1147/sj.263.0276
- Zambonelli F (2016) Towards a General Software Engineering Methodology for the Internet of Things
- Zhang C, Shi X, Chen D (2014) Safety analysis and optimization for networked avionics system. In: 2014, IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC). IEEE, pp 4C1-1-4C1-12