Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
COPPE UFRJ

SCENARI$_{oT}$
SUPPORT FOR SCENARIO SPECIFICATION OF INTERNET OF THINGS-BASED SOFTWARE SYSTEMS

Valéria Martins da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Guilherme Horta Travassos

Rio de Janeiro
Abril de 2019

SCENARIoT

SUPPORT FOR SCENARIO SPECIFICATION OF INTERNET OF THINGS-BASED SOFTWARE SYSTEMS

Valéria Martins da Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

____
Prof. Guilherme Horta Travassos, D.Sc.

____
Profª. Flávia Coimbra Delicato, D.Sc.

____
Prof. Ivan do Carmo Machado, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2019

Silva, Valéria Martins da

ScenarIoT: Support for Scenario Specification of Internet of Things-based Software Systems/Valéria Martins da Silva. – Rio de Janeiro: UFRJ/COPPE, 2019.

XV, 111 p.: il.; 29,7 cm.

Orientador: Guilherme Horta Travassos

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2019.

Referências Bibliográficas: p. 104-110.

1. Software Engineering. 2. Internet of Things. 3. IoT. 4. Software Systems Specification. 5. Requirements Engineering. 6. Scenario Specification. 7. IoT Interaction Arrangements. I. Travassos, Guilherme Horta. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

"Hucusque auxiliatus est
nobis Dominus"

(Liber Primus Samuelis 7:12)

Agradecimentos

Agradeço a Deus.

Ao meu pai Gelcí e minha mãe Dionísia por todo o apoio, motivação, confiança. Obrigada, pai e mãe, por serem o meu porto seguro, o lugar onde posso me abrigar nos momentos de cansaço, insegurança, nas alegrias e vitórias. Agradeço minha irmã Karine e meu irmão Diego pela motivação constante e amizade genuína. Também, ao meu cunhado João Paulo pelo companheirismo e à minha sobrinha Emanuela (Leléla) pelo amor sem reservas. Família é tudo!

Agradeço ao amigo Itamar Júnior pelo companheirismo e ao professor Patrick Letouzé pelos ensinamentos no período de graduação na UFT, que foram essenciais para a continuação da minha carreira acadêmica. Agradeço aos demais amigos de Palmas e do Rio de Janeiro pela motivação constante e compreensão nos momentos que me fiz ausente.

Obrigada aos colegas do grupo ESE pelas discussões valiosas e colaboração inigualável. À Rebeca, Luciana, Talita, Bruno, Breno Bernard, Hélvio, Alessandro, Andrea, Danyllo e Victor Machado pela parceria nos trabalhos que desenvolvemos. Aprendi muito com vocês! Agradeço a Luciana, Talita, Victor Vidigal e demais envolvidos pelo suporte no momento de minha mudança para o Rio. Ao Jobson e Paulo pelas conversas e questionamentos sempre construtivos. Ao Fábio e Wladmir por todas as observações construtivas sobre o meu trabalho. Meu agradecimento ao Thiago Silva por viabilizar a execução de um dos estudos da minha pesquisa e demais colaborações. Ao Hilmer pela amizade e parceria que vou levar para a vida. Por fim, todos os membros deste notável grupo que pude fazer parte.

Agradeço às professoras Cláudia Werner, Ana Regina Cavalcanti, professores Guilherme Travassos, Breno Bernard, Toacy Cavalcante e Jano Moreira por me possibilitarem tamanho aprendizado durante as disciplinas que cursei no PESC. À professora Flávia Coimbra Delicato e professor Ivan Machado por aceitarem participar da minha banca de defesa e pelas contribuições preciosas. Agradeço aos colaboradores da secretaria do programa PESC por todo o suporte durante o curso, principalmente na finalização deste.

Por fim, meu agradecimento ao professor Guilherme Horta Travassos por me receber em seu grupo de pesquisa, por aceitar me orientar no mestrado; por todo o apoio desde minha mudança para o Rio de Janeiro. Obrigada, professor, por sua atenção, disponibilidade, motivação, paciência e preciosa orientação ao longo da pesquisa. Obrigada por todos os momentos de motivação - em especial nesta reta final - quando nos fala: - "*Be the force with you!*".

À UFRJ / COPPE / PESC, à Capes e FAPERJ pela oportunidade e apoio.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SCENARI$_{oT}$
SUPORTE PARA A ESPECIFICAÇÃO DE CENÁRIOS DE SISTEMAS DE SOFTWARE BASEADOS EM INTERNET DAS COISAS

Valéria Martins da Silva

Abril/2019

Orientador: Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

Internet of Things (IoT) é um paradigma que permite compor sistemas a partir de objetos equipados com comportamentos de identificação, sensoriamento ou atuação além de capacidades de processamento, os quais podem se comunicar e cooperar para alcançar objetivos. Assim como toda mudança de paradigma, IoT faz emergir desafios relacionados a diversas áreas de pesquisa, incluindo a Engenharia de Software nas diferentes fases de desenvolvimento.

Considerando fases iniciais de projeto, este trabalho propõe a abordagem ScenarI$_{oT}$ visando apoiar a especificação de cenários no desenvolvimento de sistemas de software baseados em IoT. Esta abordagem é fundamentada em Arranjos de Interação, os quais representam fluxos recorrentes de interação entre elementos abstratos do domínio IoT. Estes arranjos puderam ser desenhados aplicando-se uma abordagem baseada em evidência, ou seja, considerando os resultados obtidos por meio de uma revisão estruturada da literatura conduzida com o objetivo de identificar, analisar e interpretar os conceitos e propriedades do domínio IoT. A abordagem ScenarI$_{oT}$ foi aplicada em dois projetos de turmas de graduação com o objetivo de observar a sua utilidade. Os resultados dos estudos mostraram que a abordagem é útil considerando o contexto onde foi aplicada.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

SCENARI$_{oT}$

SUPPORT FOR SCENARIO SPECIFICATION OF INTERNET OF THINGS-BASED SOFTWARE SYSTEMS

Valéria Martins da Silva

April/2019

Advisor: Guilherme Horta Travassos

Department: Systems Engineering and Computer Science

Internet of Things (IoT) is a paradigm that allows composing systems from objects equipped with identification, sensing or acting behaviors in addition to processing capabilities, which can communicate and cooperate to achieve objectives. Similar to every paradigm shift, IoT brings up challenges related to several areas of research, including Software Engineering at different stages of development.

This dissertation proposes the ScenarI$_{oT}$ approach aiming to support specification of scenarios in the development of software systems that are based on the Internet of Things (IoT) background. This approach is grounded on Interaction Arrangements which represent recurrent flows of interaction between abstract elements of the IoT domain. These arrangements could be designed by applying an evidence-based approach, i.e., considering the results obtained through a structured literature review conducted with the objective of identifying, analyzing and interpreting the concepts and properties of the IoT domain. The ScenarI$_{oT}$ approach was applied in two projects of undergraduate classes in order to observe its usefulness. The results of the studies showed that the approach is useful considering the context in which it was applied.

Contents

List of Figures

List of Tables

Abbreviations

EbSE	Evidence-based Software Engineering
ESE	Experimental Software Engineering
IoT	Internet of Things
IoT-bSS	IoT-based Software Systems
S-bRE	Scenario-based Requirements Engineering
SbA	Scenario-based Approaches
StLR	Structured Literature Review
SSE	Software Systems Engineering
VAS	Visual Analog Scale

1 Introduction

This chapter aims to present the context, motivation, and problem targeted in this research. Also, we explain the objectives and the methodology adopted to achieve them, as well as the organization of this Master dissertation.

1.1 Introduction

In 1999, at a lecture to Proctor & Gamble Company, Kevin Ashton from MIT's Auto-ID Lab coined the term the "Internet of Things" (IoT). At that moment, Ashton meant that information from things should be gathered as they are "*the basis of our economy, society, and survival.*" However, this sort of data cannot be gathered from human beings by pressing buttons, typing, taking pictures or performing laborious tasks such as scanning barcodes. Computers would have to gather information from things that could "see," "hear," "smell" and "touch" the world from themselves. In a broad sense, the IoT represents the possibility of giving some "smartness" to everyday objects adding to them some capacities to narrow the gap between the virtual and real worlds. It represents a new paradigm revolution regarding software systems.

Every paradigm changing can impact and bring new challenges for the involved field. According to Li *et al.* [51], "*when millions even billions of things can be integrated seamlessly and effective, IoT can be applied widely in numerous areas*," but achieving it is a challenge. The full realization of the IoT depends on and brings challenges for several research areas, including Network, Security, Privacy, Power consumption, Big Data, and also Software Systems Engineering (SSE). Since the term was coined, the IoT is supposed to affect society in many ways, including the manner software systems are engineered and developed, also impacting the essential manner that individuals use and interact with them. The equivalent occurred, for instance, when the Web had been in the spotlight as one of the major platforms to implement information systems, affecting systems engineering and development compared to other software platforms.

1.2 Context

The European Commission has foreseen that by 2020 there will be 50 to 100 billion of those "smart devices" connected to the Internet [78]. This availability of devices will enable creating a myriad of software systems limited only by imagination, corroborating with arguments that the IoT will pose significant challenges to software development as software will be everywhere. It leads to the need for new answers to questionings such as:

- How to deal with the high growth of distribution and heterogeneity?
- How to scale software systems (users, objects, infra, etc.)?
- How to allow the least degree of security and privacy?
- How to build software systems composed of humans, services and "things", and how to manage interactions among “things” to achieve opportunistic interactions based on humans’ behavior and movements?
- Who is the customer? How to specify requirements?
- How to identify and deal with conflicting requirements when the various components of the IoT system are supposed to be concerned with distinct domains and developed by different stakeholders?
- Will the development of IoT-based Software Systems (IoT-bSS) need new approaches to gather and communicate information as the development process is supposed to involve stakeholders from different areas?
- Will the development process be least “be-spoken” and more focused on creating and developing solutions together with stakeholders adding the need to apply ideation methods?

Those software-related questionings are still in a high level of discussion and need investigation, but one can summarize them: should existing software engineering methods and development approaches be tailored or new ones proposed in order to fit issues involved in the IoT realm properly?

Research combining IoT and SSE is ongoing. Around the years 2016 and 2017, some authors [49,77,81,92,93] have pointed out roadmaps and possible software challenges to face when developing IoT-bSS. Such authors argue that conventional software engineering approaches and technologies may not be feasible for this contemporary paradigm, needing research from requirements engineering to the following software development cycle activities. “*Past software engineering techniques can be harnessed and adapted to the challenges of today’s IoT*” [49]. Such context reflects that there are many challenges behind engineering IoT-bSS – many still unaware – besides the lack of answers, but first steps have been arising toward addressing the known issues so far.

1.3 Motivation

This work gives a step further towards the research combining IoT and SSE, which is part of the Contemporary Software Systems research agenda of the Experimental Software Engineering (ESE) group at COPPE/UFRJ. Such an agenda is an umbrella that has been accommodating research on topics such as Ubiquitous and Context-Aware systems, Interoperable systems, and recently, IoT and Internet of Everything. A CNPq Project called CAcTUS[1] – Context-Aware Testing for Ubiquitous Systems – is one of the results of the research agenda, in which software engineering investigators established particular software testing approaches and research questions regarding the interoperability, quality, and usability of context-aware software systems.

Besides, other ESE group collaboration opportunities have been raised in conjunction with companies and other research institutions such as Fiocruz – Fundação Oswaldo Cruz. The project developed with researchers from Fiocruz aims at supporting neglected tropical disease diagnosis processes through software and IoT technologies. The project counted on a multidisciplinary team, where software engineers of the ESE group were responsible for the entire software development lifecycle activities.

At the teaching level, undergraduate classroom projects have also been mentored with an emphasis on IoT technologies. Some of these classroom projects are concerned with the monitoring of the quality of air, supporting newborn children health care, and also monitoring a shrimp farm, which is considered to perform one of the observational studies for the development of this research, as it will be shown in Chapter 6.

All these software system projects enabled observing and analyzing the application of approaches, methods, and software technologies conceived at the ESE group, besides raising research opportunities in IoT. The theme of this dissertation resulted from working in some of these projects and observing the challenges of capturing and specifying software scenarios involving the IoT paradigm.

Therefore, this work attempts at taking another step concerned to the early-phases of IoT-bSS development, focusing on the capture and specification of its behaviors involving IoT. This theme was defined once observing a lack of software technologies for that purpose in the technical literature, which was strengthened by some challenges faced by software engineers in the projects. These projects revealed some challenges, as presented below:

a) The system was supposed to operate, taking the environment changes into account. It implies building a context-aware software system.

[1] http://lens.cos.ufrj.br/cactus/

Challenge: Documenting software systems' behavior when its functionalities can be affected by the variation of context.

b) The software system should be composed of functional units based on IoT technologies (e.g., sensors and actuators) to provide ancillary services supporting context-awareness. It includes monitoring and tracking users' behaviors and the environment so as the software system do not depend on humans' manual input in specific scenarios.

   Challenge: Documenting software systems' behaviors considering the existence of "things" collaborating (whose things' behaviors were not deeply known at that time), besides the fact software system's actions might not be triggered by humans' manual intervention in given scenarios.

c) A multidisciplinary project involving stakeholders from areas such as Microbiology, Parasitology, Epidemiology, Artificial Intelligence, and Software Engineering.

   Challenge: Communicating software systems' behaviors so that stakeholders from different areas and knowledge levels could understand and contribute to the development.

Three main issues summarize the challenges raised from these projects' peculiarities, whose issues have therefore been used as drivers for the present research. These three issues are mainly concerned with the need for a deeper understanding of the IoT concept and properties, besides documenting and communicating IoT-bSS behaviors in early phases of development. The need for **understanding the IoT** is due to the impossibility of finding structured knowledge on the IoT concept and its properties at the time this research was in its beginning, being this knowledge essential to enable addressing the following two issues and future ones. The **documenting** issue is due to the lack of evidence on the feasibility of conventional techniques for tasks of documenting or specifying software systems' behaviors considering the IoT paradigm properties. Additionally, the **communication** issue relates to the multidisciplinary project nature, where the software systems' behaviors should be communicated to stakeholders and developers, some of them with no knowledge of IoT and its capabilities.

1.4 Problem and Research Question

In an attempt to address those mentioned issues, the **description of scenarios** has been appointed to support activities in the early phases of an IoT-bSS development. That appointment is justified by the background of tailoring, enhancement and reusing of scenario-based approaches, besides taking into consideration evidence on Scenario-based Requirements Engineering (S-bRE) for the elicitation, documentation, communication, and validation of requirements [70,75,80].

The proposal of some existing scenario-based approach or to tailor one to fit the IoT realm depends strongly on initial research regarding IoT conceptualization. For this reason, an effort to aggregate and summarize knowledge on the IoT concepts and properties is essential, so as "*to provide how current best evidence from research can be integrated with practical experience*" [19]. The results of this review are expected to provide an awareness related to the properties of the IoT domain that should be captured in the scenario descriptions.

Therefore, this research is based on the problem of whether scenarios can be adopted as a means to specify IoT-bSS' behaviors as long as the properties of the IoT domain are captured. In order to observe whether that expectation is acknowledged, the following research question was derived:

How to perform scenario specification of IoT-based software systems so that the IoT domain components and properties are captured?

This research question drives this dissertation, and all the actions to achieve the objectives stated subsequently.

1.5 Objectives

Considering the motivations, problem and the research question stated to drive this research **the broad objective is to take a step further toward supporting the specification of IoT-based systems having scenario descriptions as a basis so that the IoT domain components and properties are captured**.

This broad objective can be broken into two specific ones listed below:

A. **IoT concepts and properties review**: an examination of the technical literature to gather and summarize the IoT domain concepts and properties so that the results can serve as input not only to achieve the broad objective of this research but also to support future ones
B. **Software Technology to support IoT based software systems specification**: development and refinement of a scenario-based approach to support the specification of IoT-bSS so that the properties of the paradigm are properties captured and documented.

1.6 Methodology

A research methodology has been designed in other to implement this work considering its objectives. The methodology (Figure 1) is divided into three main steps:

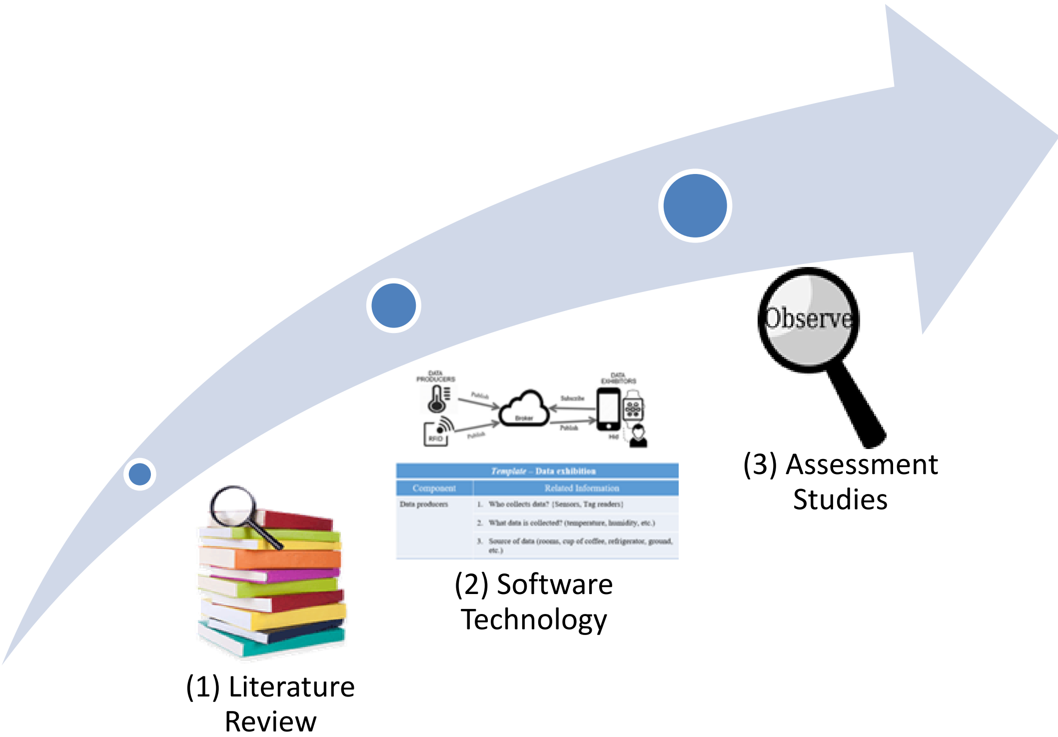


Figure 1 - Research methodology

(1) Literature review, (2) Software technology proposal, and (3) Assessment studies. Step (1) is related to the objective **A** explained previously, whereas steps (2) and (3) have a concern to achieve the objective **B**. Each step is briefly described as follows.

A. **Literature review:** performed during 2017 with an update in December 2018 to cover new publications made available between 2017 and 2018. The purpose of this step is to review the technical literature more systematically, to raise a more in-depth IoT understanding and its challenges, identifying its definitions, characteristics and current areas of use. This review is detailed in Chapter 3 and aimed primarily at bringing inputs to enable the technology proposal of this research, but also at contributing to further research that needs a more structured knowledge on IoT.
B. **Software Technology proposal:** once obtained the results from the literature review, a work composed of two subsequent steps was carried out toward proposing a technology. In the first step (starting in the second half of 2017) occurred a process of identifying IoT Interaction Arrangements, which are recurrent interactions among IoT domain's elements. This process and its results are described in Chapter 4. Further, the resulting arrangements enabled deriving information considered relevant to scenario description, whose information has

been compiled into catalogs (first half of 2018), which are the core of the technology proposal (Chapter 5).

C. **Assessment studies:** the proposed technology was examined employing two empirical studies, both regarding IoT-related projects of undergraduate disciplines from distinct universities in Rio de Janeiro. The first one was performed between April and August 2018 aiming at observing the application of the arrangements and the information catalogs in specification activities. The second study took place between June, and July 2018, in which the interaction arrangements were the principal object of study as their utility in other development activities (not only for scenario specification) was a matter for conjecture. The results from these studies were expected to support refining and improving the software technology, and also to raise issues for further research. The assessment studies are detailed in Chapter 6.

1.7 Organization

This dissertation is laid out as follows:

- **Chapter 1 – Introduction:** this first chapter introduces this work, presenting motivations and drivers for this research, as well as the problem addressed, the objectives and research methodology applied.
- **Chapter 2 – Theoretical Background:** the second chapter introduces the theoretical background of this work, which is summed up in IoT and Scenario usage in the context of software development.
- **Chapter 3 – Literature Review:** In Chapter 3, it is described the literature review focused on IoT. A process of gathering and analysis was performed with emphasis on secondary research addressing concepts, properties and applications areas of the IoT realm.
- **Chapter 4 – IoT Interaction Arrangements proposal:** It has been identified and described the need for raising recurrent interactions among *things* in IoT, which has been called IoT Interaction Arrangements. The process of identifying and analyzing these arrangements is described in Chapter 4, where real applications are mentioned in order to illustrate the arrangements instantiation.
- **Chapter 5 – Software Technology proposal:** we propose Catalogs of information to support scenario specification in early phases of software systems development, being grounded in the IoT Interaction Arrangements.

- **Chapter 6 – Assessment Studies:** describe two observational studies carried out to investigate the usage of IoT Interaction Arrangements and the Catalogs for scenario specification.
- **Chapter 7 – Conclusions:** finally, conclusions are presented as well a summary of the main contributions of this work and some limitations, besides outlining issues and possible paths for further research.

2 Theoretical Background

This chapter aims at introducing the main concepts that are in the orbit of this dissertation. Scenarios are explained from the software engineering lens as well as the role that scenario descriptions can play considering early phases of software development. Scenario descriptions are also discussed as a possible instrument to help practitioners in the process of understanding and documenting contemporaneous software systems objectives, specifically with a basis on IoT, which is also introduced in this chapter and discussed more deeply in the next chapter through a literature review.

2.1 Scenarios in Systems Requirements Engineering

2.1.1 Introduction

An important measure to determine whether some software system (or any product) achieved success is related to the level in which that software met the purpose for which it was intended [63]. However, understanding the purposes and stating what a system must do is considered a hard activity.

The "what" a system must do it is related to its functional aspect, whereas the "how" relates to the form functions might be provided considering the properties/constraints of the system. These two distinctions are typically known as "system's requirements." In other words, as defined in the IEEE-STD-1220-1998 [96], a requirement is:

> *(…) a statement that identifies a product or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for a product or process acceptability (by consumers or internal quality assurance guidelines).*

Software requirements are often regarded to be treated in early phases of development, in activities that generally involves identifying the needs of stakeholders and documenting them. The form each activity is performed vary (or should vary) from projects, as well as their resulting artifacts. For instance, activities to identify the "what" a system is intended to do may be distinct from "bespoke" to "off-the-shelf" software projects if it is considered that to build the former one is necessary an active involvement of stakeholders in order to adequately capture their specific needs, whereas off-the-shelf ones may not, due to its generic nature. Further, the form by which requirements are documented should vary from less to more formal depending on each project, or which would consume those documents, etc. Those activities and all challenges involved in are part of a broader discipline of Requirements Engineering (RE). Hull *et al.* [43] defined RE, and it has been used as a baseline in this research:

> *[Requirements engineering is] the subset of systems engineering concerned with discovering, developing, tracing, analyzing, qualifying, communicating and managing requirements that define the system at successive levels of abstraction.*

In this definition, it is possible to notice some activities that permeate the treatment of requirements, and some of them may extend to the entire development life cycle. Many works evidence the importance and impact of RE in software development. Bell and Thayer [13] have evidenced by employing empirical research the importance of software requirements and the incidence of problems across projects, describing that "the requirements for a system, in enough detail for its development, do not arise naturally. Instead, they need to be engineered and have continuing review and revision". It is by Brooks [18] which claims that "the hardest single part of building a software system is deciding precisely what to build."

Latest works have also investigated the importance of RE. Fernández [29] evidenced from data of internationally distributed surveys that the most frequent problems experienced to lead to project failure are related to software requirements. Results show that incomplete requirements are the most frequently stated issue, followed by communication flows between teams and customers, besides underspecified requirements. These results are similar to data drawn from surveys conducted by the Standish Group in 1995 and 1996, as reminded by Hull *et al.* [43], whose results evidence that "Incomplete requirements" was the most reported reason for project failure (13.1%), and "Clear statement of requirements" being the third one for project success (13%). These studies highlight the importance of engineering systems' requirements so that they can adequately be captured, documented and consumed. That underlies the statement that "nobody can refute the importance of RE and its challenges" [29].

The challenges on RE discipline are concerned, e.g., to how orienting the practitioners to specify requirements, how to communicate them, or even dealing with terminologies and developing applicable methods, templates, etc. Software Systems' behaviors are commonly captured from requirements specification that has a tradition to be written by "shall" statements as separate individual elements. This sort of specification is commonly related to "heavy" processes that emphasize up-front modeling. It may be considered hard to write, interpret and guarantee its quality in the sense of preventing, e.g., ambiguity and omission information. Further, it has been discussed that this conventional manner of documenting systems' behaviors may be time-consuming in such level that makes it unfeasible for projects with concurrent time-to-market and crucial need for early/continuous delivery of "valuable" software; or even innovation-related projects with a high basis on ideation or that ones with possibly not well-defined or volatile goals.

Considering these and other pain points when dealing with software systems development from specific conventional manners - not just from the requirements spectrum - Alexander and Maiden [3] have pointed out what they called a "modern basis for system development." Bringing to the focus of this work, according to the authors there exist some alternatives to specify the "what" a system is intended to do, not only by specifying requirements from "shall" statements. One of those alternatives has a basis on specifying **scenarios**, which can be a remedy for the complexity and/or a complement to assist understanding and documenting systems' behaviors.

2.1.2 Definition

According to the Merriam-Webster dictionary, a Scenario is "a sequence of events especially when imagined"[2]. The term has been widely addressed and interpreted from the lens of diverse fields such as Systems Simulation [9,41,50,73], Human-Computer Interaction [20,62,79,84], Agent-based systems [15,48,59,61,94] and also Requirements Engineering in particular concerned to requirements analysis and specification [25,34,42,49,70,80,86,95,96].

The definition of scenario may vary according to each field. In the context of software systems disciplines, one of the seminal works on the scenario concern describes it as "*a description of an activity in a narrative form; a description of a set of users, a work context, and a set of tasks that users perform or want to perform*" [62]. Slightly more recent, the work from Liu [52] describes the scenario as a "*temporal sequence of interaction events between the intended software and its environment (composed of other systems and humans).*" Also, Glinz [33] refers to a scenario like the following:

> *"[Scenario] is an ordered set of interactions between partners, usually between a system and a set of actors external to the system."*

[2] "Scenario." *Merriam-Webster.com*. 2018. https://www.merriam-webster.com/dictionary/scenario (28 October 2018).

In general, a scenario is a temporal sequence of actions; an ordered set of interactions among parts. To avoid confusion, when using the term **scenario** in this work, it will refer to a statement of some temporal sequence of actions. An example of a scenario is "The fridge buys missing meat from the e-commerce service." That being said, it needs to be distinguished from the **scenario description,** which is going to be frequently used in this work. Whereas the scenario will refer to a statement, a scenario description will refer to the detailing/description of the scenario statement. Following the previous example, its description could be:

"The fridge identifies there is no meat for the lunchtime. The fridge requests the product from the e-commerce service. The delivery robot delivers the product".

The scenario above was described in natural language, but the format is relative. Scenarios can come in a variety of shapes and forms, depending on the field and other variables (just as in its definitions). Many are the possibilities of styles or forms it can be described, or even what composes its filling (the content), as well as other diverse possibilities from different views to characterize scenarios or scenario usage approaches - the so-called Scenario-based Approaches (SbA).

2.1.3 Characteristics

The work from Rolland *et al.* [71] presents a 4-dimensional framework in other to assist classifying scenarios along with four different views: the *form*, *contents*, *purpose*, and *life cycle*, as depicted in Figure 2. The authors conducted this work motivated (a) *to help to understand and clarifying SbAs*, (b*) to situate the industrial practice of scenarios and* (c) *to assist researchers in developing more innovative SbAs*. The four views of this framework will guide the matter of this section in introducing and clarifying scenarios considering the systems requirements field (related to (a)). Also, the framework was used as a basis to designate the specific view to be focused on this research investigation (related to (b)).

A scenario is expressed under a specific *form*. There exist several styles or description levels to describe or represent scenarios: from natural language, structured text, customer journeys, storyboards, diagrams, pictures, animation, maps, wireframes, mockups, prototypes, and many other [35,52,80]. Some forms to write scenarios have been studied with regard to their advantages and disadvantages, including forms such as narrative text, step-by-step description in natural language (proposed by Cockburn [23]) and also other representations in different levels of formality such as flow diagrams, message sequence charts, statecharts, regular languages, etc.

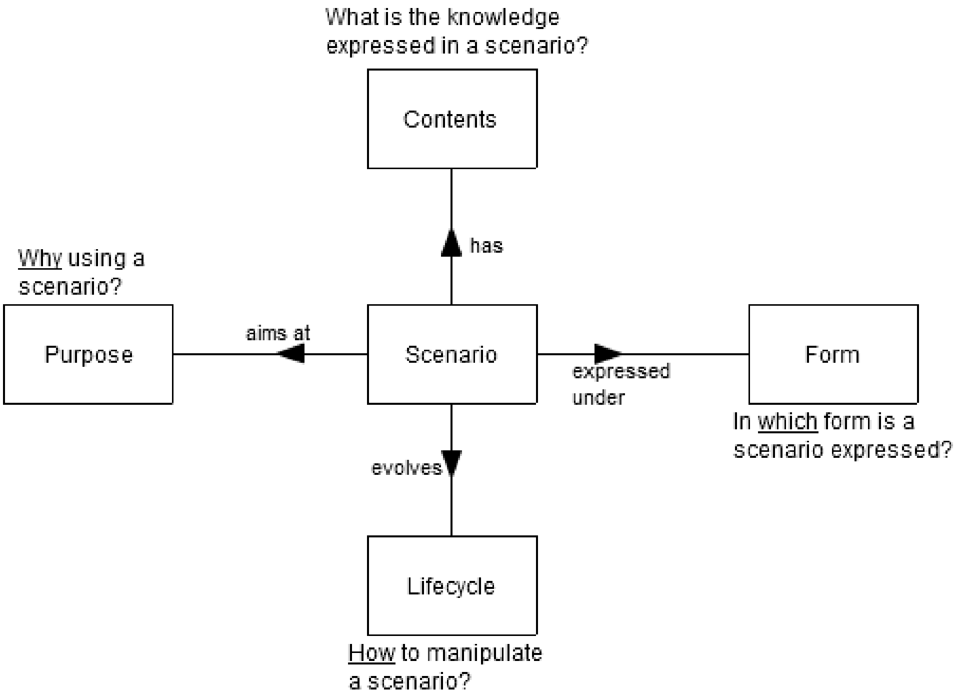


Figure 2 – The four views on scenario (extracted from Rolland (1998))

Narrative texts are considered convenient for both writing and reading, easy to understand for domain experts, analysts and systems developers [12], but it brings some disadvantages due to possible imprecision in describing the sequence of actions clearly or in distinguishing user actions and system responses. Instead, the Cockburn style exhibits a precise sequence of actions but do not expresses how to treat existing alternatives in the flow systematically. Also, it may not make clear the boundaries of the system and the environment. On the other hand, although more formal descriptions such as diagrams, charts, and regular languages can be more precise, they may impose disadvantages such as the high effort in writing, besides expenditures related to readability [33]. Multimedia scenarios, mock-ups, and others can be alternatives to text-based forms as sounds and pictures are more informative than the textual, but building that kind of representations might be a laborious and time-consuming process [39]. The decision of which description form to adopt depends on the role (s) a scenario will play in the systems engineering process. Therefore, it is strongly related to intended usage purposes.

The *purpose* view leads us to the question "Why using a scenario?". Scenarios can be used as a **complementing approach or even as the key one in some activities of software development**. For instance, Regnell and Runeson [69] combine scenarios from RE with different areas of Verification and Validation (V&V) motivated to "minimize modeling effort by utilizing the same information for several purposes," promoting traceability from requirements to test activities. They also consider results from a survey of industrial software projects indicating the need to base system tests on scenarios. Also, Whittle and Krüger [89] argue that scenarios are a straightforward manner of beginning the process of software requirements elicitation or test case development.

Liu and Yu [52] emphasized the use of scenarios for eliciting or validating requirements or being a basis for test cases writing. Further, as scenarios provide “the snapshots of possible design solutions or fragments of solutions, this concreteness facilitates the communication process between stakeholders and implementers of the system-to-be. It is by the research report from the Cooperative Requirements Engineering With Scenarios (CREWS), a long-term research project that has contributed significantly to the area. They reported that scenarios are useful for acquiring and validating requirements due to their informality, being then useful for communication, explanation, and negotiation [55]. It is also consistent with Leite *et al.* [66] that indicates scenario descriptions as a means to elicit application knowledge as well as to register elicited information. Go, and Caroll [35] states that scenario-based system design provides shared vocabulary among the people participating in the system development, envisions the uncertain future tasks of the system users and turns more natural developing instructional matters. In summary, scenarios or SbAs can be used from earlier to late software project activities, being evident the usage in requirements elicitation, analysis, validation, requirements explanation and communication, test case development, etc.

Considering the diversity of utility and the opportunity of reusing scenarios through activities, when scenarios are treated as artifacts they are created, transformed and deleted over the development process. It is the concern of the *lifecycle* view, questioning “How to manipulate a scenario?”. This view classifies scenarios as living artifacts having a transient or a persistent *lifespan*. Transient scenarios are meant to be a support for some RE or design matters, being disposed of after their usage, whereas persistent ones have a long lifespan due to its use as an active part of requirements specification or when project documentation keeps track of them. Especially the persistent ones, scenarios suffer operations throughout their lifecycle, that is, they are generated (from scratch or by reuse) and can be refined, integrated, expanded or deleted. For some of these operations on scenario artifacts, the contents must be aligned to the purposes so that scenarios can achieve the goals of the role to be performed.

Scenarios have no defined content if they have no purpose. From the definitions, it is possible to perceive that contents might be conducted from a temporal sequence, but with no defined purpose scenarios tend to be open-ended, too wide or too narrow, containing too much noise and no relevant information at the end. For scenarios classification considering the *content* view, they can be expressed through four *facets* (a term introduced by Rolland (1998)) or viewpoints: the *abstraction*, *context*, *argumentation,* and its *coverage*, each one strongly dependent on purposes. The abstraction facet is related to the degree of concreteness of a scenario. In one hand, *Instance scenarios* are those described with little or no abstraction, concentrating on details and actual argument values. This sort of scenario can be considered useful for activities of requirements elicitation, as people react to real things and as it reduces ambiguities [65].

On the other hand, *Type scenarios* describe facts in categories, with little concreteness, e.g., actors are not expressed with specific names such as Paul, but from categories such as Consumer. Besides differing scenarios from their abstraction degree, they can differ according to the amount of contextual information they capture. The *context* captured can be limited to, e.g., information about the internal system behavior, organizational information, the interaction between the system and the environment, etc. Besides the context, capturing issues, decisions, positions, and arguments can also occur in scenario description, which is the concern of the *argumentation* facet. That is, scenarios can capture information of problems, their alternative solutions and the arguments that ground a decision for a given solution. Finally, the *coverage* facet classifies scenarios considering whether they capture functional, non-functional or intentional aspects, in which the function can be decomposed into structure, behavior, and function and the intentional capture information of organization's objectives, intentions, and goals.

2.1.4 Challenges

Having in consideration possible issues in applying scenarios when dealing with systems engineering, whose issues may be mainly related to the diversity of aspects and facets of scenario descriptions, then it makes useful proposing materials such as guidelines in order to instruct a more systematic authoring and usage of scenarios in determined contexts. Works have raised in the sense of proposing good practices, guidelines and other efforts to deal with some pain points. Among other contributions, researchers from the CREWS project have proposed guidelines for authoring, writing and also reusing scenarios [67,68,72]. Similarly, the present work is also an attempt to provide support for scenario description, but with a focus on contemporary systems specification and concerning to guide capturing IoT-related information to fill scenario descriptions content correctly.

Besides the motivations presented in the previous chapter, the decision of providing this support is reinforced because of the lack of tools, techniques, and approaches for requirements engineering activities - especially those based on scenarios – that have a focus on contemporaneous systems behaviors, specifically IoT-based systems. That is, we have searched in the technical literature for approaches focused on IoT, but it was not possible to found from the results of our searches for (not later than 2018). That being said, it is understood that it represents a need for research. Then, a first step to take toward proposing that type of support is investigating initially on the content view of scenarios.

As mentioned above, scenarios' content can be composed of several types of information on different abstraction levels [53]. Therefore, it should be a responsibility of the in charge requirements engineer/analyst to decide what would compose the fill of scenarios considering projects' parameters. In a project immersed in a paradigm shift where many new requirements appear it is crucial to understand the paradigm underway so that approaches applied over the development process are in accordance.

Taking the description of systems behavior into account, it is no longer about expressing what happens when a form is submitted, or "*what if*" a button is clicked, or "what if" the printer is not responding, or even "what if" log-in credentials are not correct and the user wants to recover it. Software systems are no longer "hidden behind computer screens," but they are everywhere "seeing," "hearing" and "smelling" the environment, intervening and becoming involved in the real world. The fact that the environment is continually changing extends the possibilities of interacting, controlling and managing it. Therefore, correspondent systems behaviors need to be envisaged. Taking into consideration the objective of supporting scenario specification in IoT projects, it will require to deeply understand what IoT means and what its properties are so that they can be captured in the scenario description.

Broad concepts of IoT are presented as follows, but due to the lack of structured knowledge, a more systematic discussion is proposed and presented in Chapter 3, which addresses concepts and properties of IoT from the results of a structured literature review.

2.2 The Internet of Things

2.2.1 Introduction

How about one receiving their car's help plea when it has just been stolen and being notified of the car's movements? What if one's alarm clock goes off the right minutes before the estimated arriving time of a bus to the nearest stop of some one's house? Also, how about giving some "intelligence" to large-scale agriculture irrigation systems, so that it becomes able to "decide" the right moment to irrigate plantation according to variables crossing such as s*oil moisture, air humidity,* and *air temperature?* All those example applications make visible some contrasts against traditional systems that people are used. Computer systems are entering into a novel paradigm: the Internet of Things.

As it was mentioned in the introduction, the term IoT first came to attention in 1999 from Kevin Ashton speech to Proctor & Gamble Company. It has been crucial to disruptive digital innovations since the beginning of the second millennium when first attempts for improving the visibility of objects using electronic tags (e.g., RFID) went ahead. IoT represents a real promise to change the way people work, exercise, take care of health, use transportation facilities and many other people's routine activities. Besides IoT promises to impact people's lives, the full realization of it also brings several technological challenges and research opportunities when talking about contemporary software systems in many areas: Network, Privacy, Data Analysis, Power consumption, Software Systems Engineering, and many others.

2.2.2 Broad definition and characteristics

At the moment this area is gaining much attention by research initiatives and the definition have been in development. A broad definition of IoT is the following:

> *The Internet of Things is a concept in which the virtual world of information technology integrates seamlessly with the real world of things* [83].

First research initiatives based on the concept of IoT are concerned with and limited to the Radio Frequency Identification (RFID) which is an electronic tag technology [16,17,30,42,44]. Electronic tags, when coupled to a reader network, allow continuous tracking and identification of physical objects. Hu [42], for instance, present an example of a demand from Wal-Mart to tip an RFID Tag to each fresh meat to track their temperature history as they move through the supply chain. The objective of Brock in his work [16] is to create a "smart world," and it indicates the basis for the IoT, which is:

> *(...) An intelligent infrastructure linking objects, information and people through the computer network. This new infrastructure will allow the universal coordination of physical resources through remote monitoring and control by humans and machines.*

More recent work [7] considers that the basic idea of IoT 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 addressing schemes, can interact with each other and cooperate with their neighbors to reach common goals.

Although these definitions present some differences and are found in distinct abstraction levels, it is possible to perceive in a high level that in the IoT paradigm devices are diverse, and from the internet, it becomes possible to consume data from the devices' resources in order to reach goals.

Taivalsaari and Mikkonen [81] summarize the fundamental differences between IoT development and mainstream mobile and client-side web application development; some of the differences are:

- IoT devices are part of a system
- Rebootables vs. systems that never sleep.
- The number of computing units (devices/CPUs) in IoT systems is often dramatically larger
- IoT devices are embedded and often invisible.
- IoT systems are highly heterogeneous

2.2.3 Challenges

As every disruptive paradigm, the IoT brings new requirements and challenges for its realization including for the software systems engineering point of view. Systems based on IoT are intended to draw on (and control) a range of data inputs from diverse sources. That is not similar to conventional systems that rely on users triggering actions from mouse/keyboards and waiting for a systems' response. Some works have highlighted that conventional methods for engineering and developing software might need tailoring to get practical when dealing with the IoT. According to Larrucea *et al.* [49] "*past software engineering techniques can be harnessed and adapted to the challenges of IoT. Nevertheless, new approaches to standard software engineering techniques are also needed*".

Although the research on software engineering is quite recent, works from the technical literature have exposed challenges and also presented results of the first steps toward proposing solutions for determined demands. Zambonelli [93] took the first step toward a general discipline for engineering IoT systems and applications. The author sketched a methodology structured with some general guidelines and identified the different steps of the software process.

From the IoT point of view, Bassi *et al.* [11] present a reference architecture, the IoT Architectural Reference Model (IoT ARM). As a reference architecture, the authors address a range of topics:

- Functional elements and Interactions
- Information management
- Operational features
- Deployment of systems.

The authors discuss some views (representations of one or more structural architecture aspects): the Functional View, IoT Information View, and the Deployment and Operational one. The Functional View describes the functional elements, and it is composed of nine functionality groups, as depicted in Figure 3. The IoT ARM also aims at describing essential building blocks and identify design choices to deal with conflicting requirements, providing view and perspectives on distinct architectural aspects which are of concern to stakeholders.

This reference architecture has been analyzed by Cavalcante *et al.* [21], together with the *WSO2* architecture [31]. It brings contributions as presents recent proposals of reference architectures for IoT in the light of some IoT platforms requirements, concluding the proposals need to go a step further towards maturity to fulfill the essential requirements of the IoT realm.

Taking the IoT ARM as a reference for the sake of elucidation, we can highlight that the scope of this dissertation corresponds to the high-level aspect of IoT architecture views, the Functional one, and specifically the IoT ARM's Application functionality group. Some works that discuss IoT in that high-level view come in the next paragraphs.

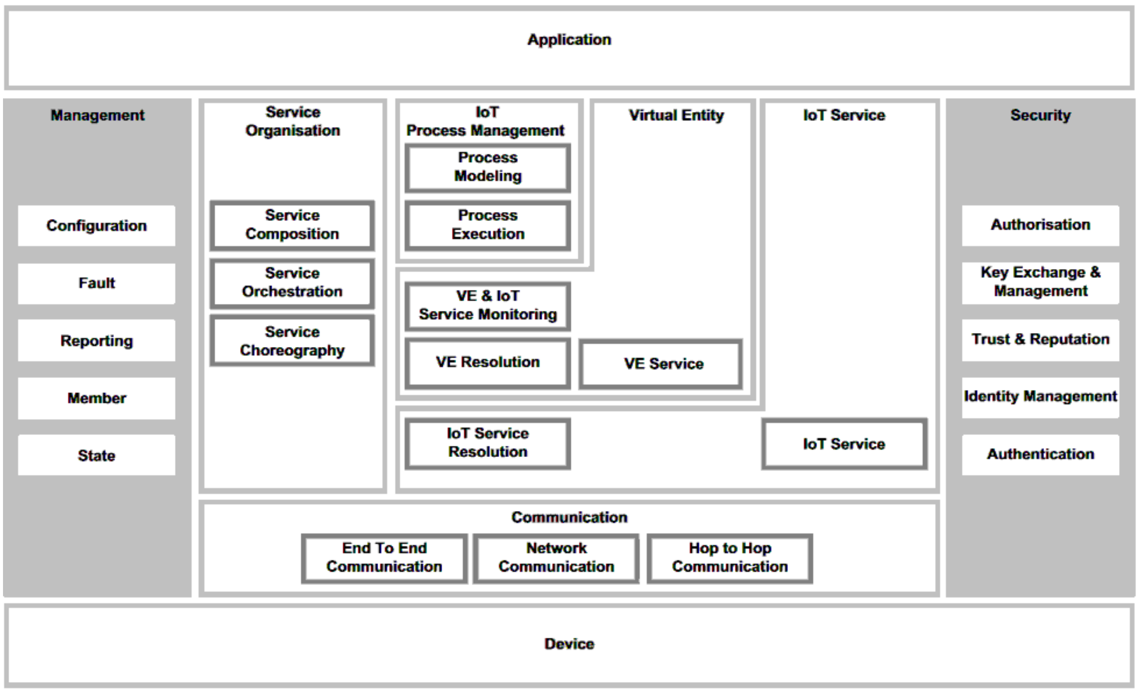


Figure 3 - Functional-decomposition viewpoint of the IoT Reference Architecture

On the concern of requirements analysis step, Zambonelli [93] highlights the activities of identifying stakeholders and users, defining functionalities framed as *policies*, *goals*, and *functions* and checking the feasibility from the infrastructural analysis. Spinellis [77] calls this paradigm shift "a maze of problems" because he considers that several challenges will arise, starting with requirements. One challenge is concerned with the clash of requirements (one interfering with each other) when multiple IoT nodes and applications get integrated. It is also highlighted that issues captured during requirements' elicitation might need to be obtained dynamically as the systems operate.

Opportunistic interactions among things are also a critical challenge in the IoT realm. Guo and Zhang [37] have addressed the problem of "information/resource sharing within and among opportunistic communities (with pairs of devices) that are formed based on the movement and opportunistic contact nature of human". They highlight that the social side of the IoT - in other words, the "*harmonious*" interaction between human and IoT - has yet been not well explored.

Another work related to the early-phase of IoT-bSS development is from Alqassem and Svetinovic [5,6] with a focus on security and privacy requirements. The authors argue that "the most complex challenge from the requirements engineering perspective is the difficulty of specifying requirements - security and privacy requirements in particular - for a system with so many components that can be randomly integrated into various systems at various times and places. Due to the diversity and complexity of the IoT, it is difficult even to envision what system an object will be a part of". The authors identified a need for a taxonomy of security and privacy requirements, and Alqassem's work [6] takes a step to make progress on requirements engineering by proposing a framework to specify privacy and security requirements in the earliest stages to provide a proper development of IoT.

The work from Morin *et al.* [58] presents an approach that addresses two primary challenges of IoT applications: (1) the distribution over an extensive range of processing nodes and (2) high heterogeneity of processing nodes and their protocols. That approach includes a modeling language, a methodology, and tools, aiming primarily to allow developers to be abstract from heterogeneous platforms and IoT devices in order to define and model the IoT systems' architecture upfront. After the architecture definition, the approach allows the specification of business logic employing statecharts in a platform-independent way. From this work, it is possible to notice some concerns about specifying systems' business logic on a defined upfront architecture considering the platforms and IoT devices available.

2.3 Chapter Considerations

This chapter presented the basic concepts on two main topics that are related to this work: Scenarios from the requirements engineering point of view and the IoT paradigm. Only broad definitions were presented so far as it is going to be more in-depth discussed in the next chapter when presenting the results of a review study on that subject.

The first topic has a history of being suggested as an alternative to the matter of specifying systems' behaviors through scenario specifications (or a complement for), and also a history of SbAs that have gone through tailoring to get suitable for some project peculiarity. Due to the multi-aspects of scenario, a research objective has been delineated which aims at supporting employing scenario descriptions in the context of IoT-bSS development, whose objective also has motivation on the lack of research addressing early phases of IoT systems development. To achieve this, we understand that an in-depth understanding of IoT is needed to ground the research, as the concepts are still nebulous and abstract.

This uncertainty explains the concern in searching for IoT properties, relying on the assumption that from framing the main properties revolving about the IoT universe (next Chapter) it could be possible to reason on how to support scenario description when developing IoT-based systems. Additionally, the IoT properties could support the emergence of arguments to be considered when constructing or tailoring future models, tools, methods or practices – for requirements engineering, development, testing, deployment, etc. - so that they are viable for the IoT context.

3 Literature Review

A Structured Literature Review (StLR) is presented in this chapter. It was conducted aiming to characterize IoT regarding its concepts, properties, and application areas IoT has been applied. The results brought a relevant knowledge to support proposing an approach for scenario description of IoT-based systems following interaction arrangements, explained in the subsequent chapters.

3.1 Introduction

Kitchenham *et al.* [19] discuss how software engineering might benefit from an evidence-based approach. The authors argue that the goal of evidence-based software engineering (EbSE) should be "to provide how current best evidence from research can be integrated with practical experience and human values in the decision of making process regarding the development and maintenance of software." The adoption of guidelines for systematic literature reviews is an essential aspect of EbSE, being a literature review a mean to evaluate and interpret relevant available research on a particular research topic, area or phenomenon of interest [27].

An StLR has been conducted in this work aiming at gaining an understanding of the IoT research domain and an overview of areas that have been applying it. Also, IoT can be considered a new field of research/development with a lack of consensus and understanding of its concepts and features, which motivates aggregating and summarizing information. The review protocol was performed following recommendations proposed by Kitchenham *et al.* [46].

Before starting a literature review, it is essential to observe its necessity. An *ad-hoc* search was carried out looking for existing secondary[3] studies on IoT. From this first search process, it was possible to find studies such as literature surveys, but most of them with no structured methodology. For this purpose, it is decided to review the technical literature more systematically, given a focus on those existing secondary studies that meet the selection criteria, even they do not mention its research protocol.

---

[3] In our perspective, "secondary studies" are the ones that survey primary studies to present a bigger picture of a domain

Some of the papers found in the *ad-hoc* search were used as control articles in the review protocol detailed as follows.

3.2 Literature Review Protocol

The protocol of this investigation has been planned by focusing on the goal of contributing to a more in-depth understanding of the Internet of Things and its challenges, identifying its definitions, characteristics and the current areas of use.

3.2.1 Research questions

The research goal has been structured based on GQM [10]:

> *To analyze* the Internet of Things *with the purpose of* characterizing *regarding* its definitions, characteristics and application areas *from the point of view of* software engineering researchers *in the context of* knowledge previously organized and presented in secondary studies regarding IoT and available in the technical literature.

From this goal, we defined the research questions:

1) (RQ1) What is the "Internet of Things"?
2) (RQ2) Which characteristics can define an IoT domain?
3) (RQ3) Which are the areas of IoT application?

3.2.2 Search string and engine

The search string has been defined considering those goals and the control papers[4] found in the *ad-hoc* search

*(("*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 ("internet of things" OR "iot"))*

[4] Atzori *et al.* [7], Bandyopadhyay and Sen [8], and Li *et al.* [51]

The search string has been applied in the Scopus[5], the engine chosen for this review as it indexes several databases of peer-reviewed sources, and as far as our experience shows, the combination of Scopus results with snowballing procedures can mitigate the eventual lack of content and provide a representative set of papers to a characterization work [56]. The search string has been applied in Scopus engine considering the title, abstract and keyword fields (*TITLE-ABS-KEY).*

3.2.3 Selection criteria and procedure

The works presented as articles shall be available on the web, retrieved from the search engine and written in English. The stated selection criteria are:

- Inclusion Criteria (IC)
    - (IC1) Provide an IoT definition AND
    - (IC2) Provide IoT properties OR
    - (IC3) Provide IoT application areas.
- Exclusion Criteria (EC):
    - (EC1) Duplicate publication/self-plagiarism OR
    - (EC2) Register of proceedings.

The selection procedure was conducted by reading the title and abstract of each retrieved study and evaluating them according to the inclusion and exclusion criteria. Two distinct readers have evaluated each study. The studies acceptance criteria occurred as follows:

- All two readers accept: The study is included.
- One reader accepts, and one is in doubt: The study is included.
- One reader accepts or is in doubt, and one reader excludes: The study is discussed.
- Two readers exclude: The study is not included.

3.2.4 Data extraction

Data extraction step aims to capture information from the selected articles to answer the proposed research questions. The data extraction form was proposed during the review planning and used throughout the process. After selecting the articles, the information was extracted according to the form presented in Table 1.

---

[5] https://www.scopus.com/

Table 1 – Data extraction table

Field	Description
Reference information	Authors, title, year and venue
Abstract	Abstract
IoT definition	Verbatim, as presented in the article (Definition research-based derived or with reference)
IoT related terms	It is associated with other definitions (ubiquitous, context-aware, pervasive, machine-to-machine, and others)
IoT application features	Characteristics of particular traits, features, properties, attributes that make IoT what it is (that achieve the IoT definition/concept)
IoT application areas	The areas (and their related applications) that will benefit from the full IoT idea deployment.
Development Strategies for IoT	The used development strategies to build IoT software (requirements analysis, design, and so on).
Type of study	It is expected to have only secondary studies, represented by Survey, SLR, others.
Study properties	Protocol, RQ, search string, selection criteria.
Challenges	Open opportunities in practice or research
Article focus	Main concerns presented in the articles (architecture, security, and others)
Things	A list of the kind of things explicitly stated in the article (coffeemaker, refrigerator, incubator, and others)

3.2.5 Execution

The review process was executed according to the following steps:

- **Step 1 - *Ad-hoc* Search.** In this trial, two researchers performed an *ad-hoc* search to identify the existence of any secondary study related to IoT. Since secondary studies have been identified, it has been decided to review the existent articles instead of relying on primary studies. From the results of this *ad-hoc* search, three articles were selected as controls; that is, a starting point for the next step once they met the selection criteria: [7,8,51].
- **Step 2 - Scopus search.** The terms of the search string have been organized based on synonyms and similar terms. Also, the search string was adjusted in order to recover the three control articles previously selected. The total of items found was

76; the search was executed at the end of May 2017, considering the papers available in the database until this date.

- **Step 3 - Title and abstract reading.** The list of 76 articles was reviewed to remove duplicates and proceedings, following exclusion criteria. The remaining articles were later read based on title and abstract and reviewed by a 3rd researcher with more experience in the research area. From this discussion, 24 articles were selected for further reading.
- **Step 4 - Full Reading**. The two researchers read the full text of the 24 articles (12 for each, with crosschecking), considering the inclusion and exclusion criteria. Seven of them met the criteria, being those finally selected.
- **Step 5 - Snowballing.** It refers to using the reference list of an article or its citations to identify additional material [90]. In this step, a Backward and Forward Snowballing Sampling has been performed, tracking down references in the seven articles selected in the previous step and their citations. The total of articles was divided so that each researcher could be responsible for performing the snowballing in part of the articles. Nineteen works were identified as candidates, and the reviewers cross-checked the articles to be included considering selection criteria. This step resulted in the inclusion of five new articles.
- **Step 6 - Review Update**. The previous five steps were carried out between March and May 2017. We decided to perform an update in December 2018 to cover new publications made available from 2017 and 2018. The same search string was re-executed in Scopus engine, and the results were analyzed following the criteria previously established. The three reviewers conducted the update repeating Steps 3 and 4 for the new recovered results and the forward snowballing (Step 5) for the selected set. This step resulted in the inclusion of three new articles.

Fifteen articles remained, composing the final set annexed in **Erro! Fonte de referência não encontrada.**. The amount of articles selected from each step is summarized in Table 2. The data extraction table is available in **Erro! Fonte de referência não encontrada.**.

3.2.6 Results

The dataset contains papers from 2010 to 2018. It is possible to observe a growing interest in the area over the years. The results show that most of the available publications on technical literature were from 2015 to 2017, considering the period of search.

Table 2 – Total of articles selected at each step of the review

Step	Number of articles selected
Scopus search	76
Title and abstract reading	24
Full Reading	7
Snowballing	5
Review Update	3
Final set	**15**

Seven topics could be perceived as essential ones addressed from selected papers. No one covers all the topics, showing that the researchers have distinct perspectives and concerns. However, together these studies provide a wealth of information to characterization. From the statistics presented below, it has become evident that Concepts and Applications are the most topics discussed:

- Concepts - Presenting discussions regarding fundamentals, definitions, and visions behind the IoT paradigm; Articles: All selected papers address this topic, except [74].
- Technology - Introducing enabling technologies and solutions to develop and deploy IoT applications. Articles: [7,8,14,36,51,54,74,82,88].
- Applications - Describing the current state of the existing solutions and the applications of different domains as well as future possibilities to be achieved by using IoT. Articles: All selected papers.
- Open Issues and Challenges – Presenting opportunities for research and development aiming to evolve IoT. Articles: [7,8,14,36,51,57,76,88].
- Architecture – Discussing possible implementations of IoT based on different architectures proposals. Articles: [8,32,54,74,76,82,88].
- Characteristics – Making specific general features and requirements of IoT. Articles: [14,32]
- Initiatives - Research organizations, industries, standardization bodies, and governments that have an interest or put some effort into IoT. Articles: [14,36,54,57].

3.2.7 Answering the research questions

In order to support the analysis phase to answer the research questions of this review, a procedure was performed based on textual analysis. It was carried out by using codes to assign concepts, properties and application areas to portions of data, identifying patterns from similarities and differences emergent from the data extracted. The part highlighted in yellow in Figure 4 represents a code assigned to an IoT concept extracted from [7]; the colored parts in Figure 5 are codes assigned to IoT properties that were extracted from [8]; in Figure 6 there are codes for IoT application areas extracted from [54].

Two researchers conducted the analysis procedure, applying to cross-check to achieve a consensus with the analysis and as a means of decreasing potential misinterpretation and bias. A third researcher reviewed the extractions and findings. This process was performed through all the data extracted and lead to the discussions of the research questions proposed, presented in the following subsections.

> 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 addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals (...)

Figure 4 – Code (yellow) assigned to an IoT concept

> Scalability, modularity, extensibility and interoperability among heterogeneous things and their environments are the key design requirements for IoT. Industry practitioners have suggested leveraging work in the semantic web to devise comprehensive and open ontologies to address the issue of semantic interoperability for dynamic binding based SOA for IoT application design and development.

Figure 5 - Codes (colored) assigned to IoT properties

> (...) the traditional IoT is formed by three layers. The bottom is perception layer, whose function is cognizing and collecting information of objects. The middle is transportation layer which consists of OFC, mobile phone networks, and fixed telephone networks, broadcasting networks, and closed IP data networks for each carrier. And finally the top is application layer, where abundant applications run. Typical applications include in this layer are smart traffic, precise agriculture, intelligent logistics, smart industry, environment protection, mining monitor, remote nursing, safety defense, smart government etc.

Figure 6 - Codes (colored) assigned to areas IoT has been applied

- **(RQ1) What is the “Internet of Things”?**

From the 15 selected papers, it was possible to extract 29 IoT definitions. In the analysis process of these 29 definitions, it was noticed that a considerable portion of them followed a pattern in their structure concerning to explain the involved actors, the requirements and the consequences of relations among actors as part of a system. That structure was considered not to limit our interpretation, but to support a more wide-ranging IoT conceptual understanding and thus finding an appropriate and updated definition to be considered in the present work and subsequent works on IoT, including from our research group. The definitions found could be organized in chronological order, enabling to observe how the concept has evolved.

> *"An intelligent infrastructure linking objects, information, and people through the computer networks, and where the RFID technology found the basis for realization."*

From the above definition stated in 2001 by [16] and cited by [14], it can be observed that the main idea is to connect objects, information, and people. It makes clear the network necessity as a way to connect components and actors of the system. In that period the realization of IoT was limited by the RFID identification technology, which represents the starting point of IoT discussions.

> *“Internet of Things as a paradigm in which computing and networking capabilities are embedded in any conceivable object. We use these capabilities to query the state of the object and to change its state if possible.”*

This other definition - defined in 2005 by [51] and cited by [8] - does not propose the use of any technology, like RFID, but includes the idea of expanding the original capabilities of an object through technology. Also, perceiving changes in the objects’ state it is only possible by given addresses to objects first, which also enables things to communicate automatically [26]. It can be considered as an evolution of concept since autonomy was not previously discussed.

This next definition is from 2008 [26], cited by [7,32]. It introduces the purpose-idea, even vaguely:

> *“A world where things can automatically communicate to computers and each other providing services to the benefit of the human kind.”*

Another one made in 2009 [38], cited by [14,88].:

> *"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 personalities and use intelligent interfaces, and are seamlessly integrated into the information network."

In this definition, the central concept of communication and integration remains, but it has been noticed the introduction of requirements such as interoperability and integration in a seamlessly way. This definition also details what are the things in IoT, as things being virtual or physical, that can have different personalities and may use different communication protocols.

"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 addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals."

It is one of the most used IoT definitions, which is from 2010 [7], cited by [36,57,76]. It can be considered as complete while taking into consideration the "actors, relations among actors, requirements and what enables" structure. It presents the vast amount and heterogeneity of actors that can engage an interaction, and a requirement to achieve that through different addressing schemes. In this case, new actors are included, and we can observe that sensing and acting are other possible behaviors that a system can possess besides identification, differing from previous definitions. Therefore, these actors can cooperate to reach some goals.

"Interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications. This is achieved by seamless large-scale sensing, data analytics and information representation using cutting-edge ubiquitous sensing and cloud computing."

Once more, sensing and acting have essential roles in IoT, as presented in this definition from 2012, defined by [36]. The vast amount of data collection and sharing among actors can be a source to compose diversified, innovative applications. This definition also makes it clear the multidisciplinary nature of IoT as there are areas that support or leverages it, such as data analytics, ubiquitous and cloud computing.

"Everyday objects can be equipped with identifying, sensing, networking and processing capabilities that will allow them to communicate with one another and with other devices and services over the Internet to achieve some useful objective (...). Every day "things" will be equipped with tracking and sensing capabilities. When this vision is fully actualized, "things" will also contain more sophisticated processing and

networking capabilities that will enable these smart objects to understand their environments and interact with people."

Once everyday things can sense the environment, they become more aware of what is around them, which characterizes context-awareness. In this definition stated in 2015 by [88], it is perceived again that the primary concern in IoT is to leverage the connection among different things to achieve a system objective. Also, the authors explain that things in the IoT context are those objects equipped with identifying, sensing, networking, and processing capabilities, whereas other definitions exemplify things as being the providers of such capabilities, that is, electronic tags, sensors, and actuators.

In our interpretation, things exist in the physical realm, such as sensors, actuators and also anything that is equipped with identification (tag reading), sensing or actuation capabilities, which excludes entities in the Internet domain (hosts, terminals, routers, among others). The things should also have communication, networking, and processing functionalities varying according to the systems requirements.

As one can notice, the capabilities of the things evolved as observed from the definitions presented. In the beginning, the things in IoT based systems were objects attached to electronic tags. These systems present the behavior of **Identification**. Subsequently, sensors and actuators joined the systems enabling the **Sensing** and **Actuation** behaviors, respectively. It means that an IoT system can have Identification, Sensing or Actuation behaviors, or a combination of these blocks as depicted in Figure 7, and each one of these behaviors brings specific requirements. The explaining of each behavior along with examples of applications to illustrate them is presented in Table 3.

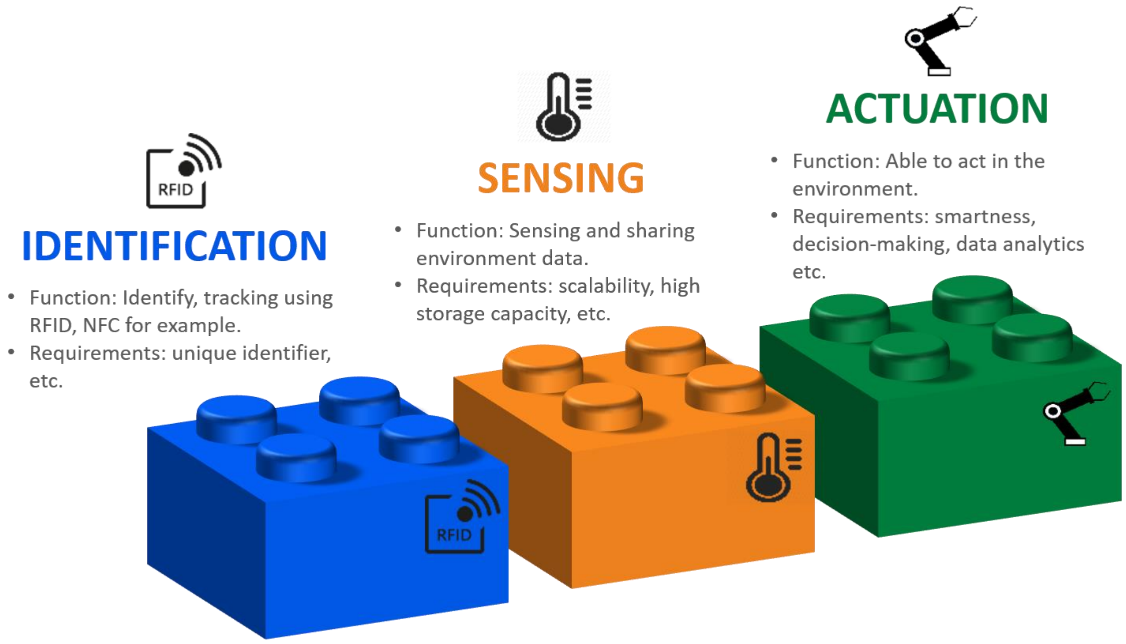


Figure 7 - IoT evolution from its three behaviors

When discussing the previous definitions, it was necessary to distinguish the meaning of “identification” referred to as objects. The reason is that an object can be identifiable in the sense of connectivity (e.g., throw IP addresses) or in the sense of physical identification when objects are tagged with electronic tags containing specific information, making it possible to identify objects through tag readers. Further, it is also relevant to elucidate the meaning of “actuation” as it may bring diverse interpretations. When focusing on the IoT context, the adequate meaning for “actuation” is precisely the one presented in Table 3. It is divergent from actions represented by methods in the object-oriented paradigm, and it is not related to objects’ processing capabilities mentioned in the IoT definition discussed previously. Actuation is exclusively related to the possibility of virtual intervening in the real world by mechanical means.

These three IoT behaviors can be found in several works as fundamental building blocks of the domain, but they are addressed in different levels of granularity or abstractions. For instance, the work developed by a team from ZTE Corporation [91] introduces the concept of “Internet of Things services,” in which a large number of applications can be included in that concept and classified according to technical features. The authors stated four types of services: (a) identity-related, (b) information aggregation, (c) collaborative-aware, and (d) ubiquitous services, in which is possible to recognize some of the three behaviors in them, but the boundaries of each service could not be clearly understood. Another work where some of those behaviors can be encountered is from [2]. In this case, instead of stating categories of services, the authors propose an abstraction based on a set of building blocks that can help to gain insights into meaning and functionality of the IoT, which are: Identification, Sensing, Communication, Computation, Services, and Semantics. This view differs from the three behaviors emphasized in this work primarily because of the Sensing block, where IoT sensors and actuators are put into the same group.

Although some works in the technical literature (including the ones presented above) address IoT behaviors in different abstractions than the trio Identification, Sensing, and Actuation, this work will be grounded on these set of three behaviors seeking to highlight the division of objects’ responsibilities as we conjecture that it can clarify and delimitate IoT solutions contributing mainly as a guide for engineering decisions (such it contributed to this research development).

To answer RQ1 from the review results, the **IoT can be defined as a paradigm that allows composing systems from uniquely addressable objects (things) equipped with identifying, sensing or actuation behaviors and processing capabilities that can communicate and cooperate to reach a goal.**

Table 3 – IoT Behaviors

Behavior	Description	Examples
Identification	The primary function is to identify *things*, by labeling and enabling them to have an identity, then recover (through reading), and broadcast information related to the *thing* and its state.	To Identify patients with electronic tags (RFID) to be detected throughout hospitals using receivers (readers) placed in departments to accelerate the identification of empty beds. Another example is the application of short-range identification technology for drug interaction and drug allergy detection. It operates by identifying patients (NFC tags integrated into their wristband) and drugs (NFC tags integrated), each tag holding a unique ID. Nurses read the patient's and drug's NFC tag by using the smartphone's NFC reader. Finally, the server verifies whether the patient is allergic to the drug or if there might be a potential interaction.
Sensing	The primary function is to sense environment information, requiring information aggregation, data processing (data treatment) and transmission. Enables awareness, thus acting as a bridge between the physical and digital world.	A new application to illustrate the capability of the sensor in the real world is from the geophysics area. Sensors have been deployed for long-distance volcanic monitoring, such as microphones and seismometers, collecting seismic and acoustic data on volcanic activity.
Actuation	Mechanical interventions in the real world according to decisions based on aggregated data or even upon actors' right trigger; relay on responses to the collected information to perform actions in the physical world and change the object state.	An example is the control of things, robots or even animals in the real world, where actuators are used in an attempt to prevent fighting between bulls in on-farm breeding paddocks by autonomously triggering stimuli such as audio warning signals or mild electrical when one bull approaches another.

- **(RQ2) Which characteristics can define an IoT domain?**

The 15 papers provided 263 excerpts, from what it has been identified 29 characteristics (**Erro! Fonte de referência não encontrada.**). One point of discussion is that the authors do not define all the characteristics presented in the articles or referred to the original work defining them. The lack of definitions hinders the research and understanding of the area since we cannot know the feature´s meaning or what the authors meant by that. Although some characteristics such as Interoperability and Scalability are told to be well defined, it is essential to establish a common understanding of the characteristics since they inspire different concepts when contextualized to distinct domains. For instance, “Efficiency” is open to many interpretations even the IoT domain is on the focus, which can be related to object’s data collection efficiency, energy-efficiency, security-efficiency, information processing efficiency as well as service adaptability-efficiency. It makes it challenging to characterize IoT and to develop more suitable solutions that meet all the desired characteristics, since they were not defined, only listed. For the same reason, it is not possible to infer that the authors are discussing the same issues, such as efficiency for instance, which from the sources can be regarding cost, size, resources or energy.

Even with this lack of definition, the characteristics pointed out are considered relevant for the characterization scenario of IoT systems. It has been retrieved the characteristics pointed out by the authors (Cited by) and the original references used by them (Reference) some references may have been used by more than one author and null (-) in case of no reference. From the characteristics we can observe that some are fundamental to an application in order to fulfill the IoT definition: “a paradigm that allows composing systems from uniquely addressable objects equipped with identifying, sensing or actuation behaviors and processing capabilities that can communicate and cooperate to reach a goal.”

- **(RQ3) Which are the areas of IoT application?**

Several application domains will leverage the Internet of Things paradigm advantages. All the application domains are only examples of areas that benefit from IoT or are supposed to do it in the future. As declared in Whitmore *et al*. “the domain of the application areas for the IoT is limited only by imagination” [88]. Despite the application scenarios were described in different levels of detail, we attempted to categorize some of them into the tree behaviors as presented in Table 4.

Atzori *et al.* [7] describe five domains: (A) Transportation and logistics, (B) Healthcare, (C) Smart environment (home, office, plant), (D) Personal/social and (E) Futuristic domain (whose implementation of such applications is still too complicated). Gubbi *et al*. [36] describe (A) Personal and Home, (B) Enterprise, (C) Utilities, and (D) Mobile domain. Also, there is also a classification of the applications for Consumer (Home, Lifestyle, Healthcare, Transport) and Business (manufacturing, retail, public services, energy, transportation, agriculture, cities, and others) [82].

Those domain categorizations can be a subpart of a categorization, which grouped the applications in three primary domains [14]: (A) Industrial domain, (B) Smart city domain, and (C) Health well-being domain. They are not isolated from each other, but there is a partial overlapping since some applications are shared across the contexts. For example, tracking of products can be a demand for both Industrial and Health well-being domains.

Table 4 - IoT application areas

Behaviors	Application type
Identification	Patient triage, resource management and distribution [36]; medical equipment tracking, secure access indoor environment management, personnel tracking, bike/car/van sharing, mobile tickets, luggage management, animal tracking, fast payment, warehouse management and inventory, identification of materials and goods [14]; verifying the authenticity of aircraft, storing health records [8].
Sensing	Road condition monitoring, patient monitoring, remote personnel monitoring (health, location), sensors built into building infrastructure to guide first responders in emergencies or disaster scenarios or sensors built into infrastructure to monitor structural fatigue and other maintenance, sensing of water quality, leakage, usage and distribution, air pollution and noise monitoring, support to diagnoses, video/radar/satellite surveillance, product deterioration [14]; monitoring chronic disease using wearable vital signs sensors in body sensors [8].
Actuation	Room lighting changing, alarm systems, remote switching off electrical equipment [7], temperature and humidity control [36], irrigation control [14], muscle stimuli for paraplegic individuals [8].
Hybrid	Buildings adjusting locally to conditions while also taking into account outdoor conditions, Robot taxis that respond to real-time traffic movements of the city, and are calibrated to reduce congestion at bottlenecks in the city and to service pick-up areas that are most frequently used [7], water waste management [36], parking system, traffic management [14].

3.2.8 Threats to validity

The first threat to validity regards to the search engine chosen. Since only Scopus was used as a search engine, some relevant studies may not have been recovered. However, it has been experienced that Scopus can give reasonable coverage when performing together with snowballing procedures (backward and forward) [56].

A recurrent issue in literature reviews is related to inconsistent terminology and restrictive keywords. In order to reduce the researchers' bias, it has been searched for other reviews and observed the adopted terms to compose our search string. Data extraction and interpretation biases were softened with crosschecking between two researchers and by having a third one to revise the results. All steps of this review were peer-reviewed; any uncertainty was discussed among the readers to reduce the selection bias. Another threat to this study validity is that a quality assessment regarding the research methodology of the selected studies has not been performed due to the lack of information from the secondary reports.

3.3 Chapter Considerations

Although IoT has been widely discussed in the literature, the initial research has not returned secondary studies carried out systematically, nor have they presented the methodology followed nor the research questions that the papers intended to answer. Except in [88], which presents some study properties, the papers found before the review update do not present a research protocol or specific methodological properties (research questions, search strings, search engines, selection criteria, articles selected, amongst others).

From the initial search results, it was noticed a need to provide research data based on sound scientific methodology. Despite the evolution and enthusiasm that new generation technologies can provide with latest tendencies including IoT, the lack of scientific rigor it is still one of the significant challenges to strengthen the basis of Software Engineering knowledge [4]. The investigation was conducted by following established guidelines and in a protocolled way, accounting for the strength of the evidence found and its replicability taking it as a concern. The questions that this review seeks to answer are aligned to characterize IoT.

From the discussion of RQ1, we 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. The idea of composing software systems from available components is not new, but one of the issues that set IoT apart is the scale at which it can be achieved and the actors involved in these new software systems. From this, shared concerns regarding the development and evaluation of such software systems should be reframed to cover the particularities of these new types of devices. A critical step towards it is to establish what quality characteristics should be contemplated. With the second research question, we moved forward in this direction.

Regarding the IoT characteristics (RQ2), from the technical literature, we recovered 29 different attributes, from which this paper discussed nine of them with clear evidence from the sources of information. Considering that the results retrieved are from secondary studies, the characteristics represented reflect more than just the 15 secondary studies, but rather the whole set of primary studies involved in them which can strengthen these results. Of the most commonly cited characteristics presented are Efficiency, Interoperability, Scalability, Privacy, and Security that reassure the definition reached in the paper. Besides, the review enabled to observe which areas of application are making use of IoT (RQ3). All of these findings were related and summarized to enrich IoT paradigm comprehension.

The contribution of this work is to present an organized perspective regarding the current state-of-the-art regarding the IoT paradigm, strengthening the discussions and evolution of the field. Taking it to the broader objectives of this dissertation, the results presented in this chapter are meant to ground the technology proposal of this dissertation, that is, enable to propose support for scenario specification of IoT-bSS framing the behaviors and properties of the IoT realm.

4 IoT Interaction Arrangements Proposal

In this chapter, the process towards IoT Interaction Arrangements is elucidated. They can be considered as interaction patterns that occur in the Internet of Things context, which has been designed grounded in the concepts and properties extracted from the literature review. The defined set of IoT Interaction Arrangements is used to design the proposed approach to support scenario description in IoT projects.

4.1 Introduction

As previously discussed in chapter 3, the Internet of Things "is a paradigm which enables the composition of systems from various devices. Thus the application composition opportunities are open-ended and limited only by imagination". In this context, the concept of "Things" is that "[they] are perceived in the physical realm, such as sensors, actuators, and anything that is equipped with identification (tag reading), sensing or actuation capabilities." From it, it can be stated that despite the large amount and diversity of devices, they play finite behaviors of identification, sensing or acting. That statement is an essential drive to this research because the three IoT behaviors are high contrasting against the traditional software-related mindset, and relevant to be considered when composing, developing or engineering IoT systems.

The composition of IoT-bSS requires the entities to be orchestrated (i.e., structured, organized and managed) to turn received data into actionable information [24] and to support a business process workflow [40]. The IoT-bSS composition includes the orchestration of entities within a logical sequence of actions, meaning that the orchestration should be based on a logical dependence between the components from a functional point of view. That is, if component B depends on the output of component A, thus it is logical that these two components are orchestrated by having A as an antecedent of B in the orchestration flow.

It has been observed actual flows when reasoning about the IoT concept and the set of three behaviors. For instance, it has been noticed that does exist a dependence among things producing data and other ones consuming it. Also, things depend on other things to trigger their capabilities, either from an actor direct intervention (system or individual[6]) or even from a systems' reaction based on data (and here there is a backward flow going through data consumers preceded by data producers).

Those logical flows among things can be distinguished in IoT-bSS due to the well-defined behaviors, which naturally leads to recurrences or even relation patterns among things when orchestrating or setting them to interact with external actors (human or not). For instance, consider the description of the following scenarios:

Scenario A – To monitor heartbeat from patients with heart diseases:

> Scenario description: Heartbeat data is collected from one patient; the data is shared to be accessed from displays (devices); doctors and nurses visualize data from their display.

Scenario B – To monitor boxes as they transit through the factory:

> Scenario description: The boxes' identification is read by tag readers placed in strategic regions of a factory; the data is shared to be accessed from displays (devices); some worker visualizes data from their display.

The main point to highlight is that, although scenarios A and B differ from each other, in general, both scenarios instantiate an equivalent abstract flow of interaction among things and actors. In that flow, things collect data and then share it with display devices - smartphones, tablets, smartwatches, smart TVs - so that interested/allowed individuals can visualize information through their devices.

That being said, what if a set of interaction flows could be designed in the context of IoT-bSS? The possibility of knowing a set of recurrently implemented IoT interaction flows has raised a conjecture that it can contribute to investigating the research problem of the present work. More specifically, the scenario description activity can take advantage of those mentioned patterns as inputs, enabling analysts to visualize the equivalent flow for some scenario, resulting in more awareness of the flow's structure and the elements participating in it, so that analysts can focus on relevant information to be captured.

From that, there has been an attempt to raise a representative set of interaction flows where IoT things participate. For this purpose, a work of investigation, reasoning, and design of interaction patterns have been performed with two main steps further described in this chapter. The outputs of each step have been arranged into structures

---

[6] Living beings, mainly humans.

called **IoT Interaction Arrangements** (IIA), representing interaction flows among things and other non-IoT elements regardless of domain.

4.2 IoT Interaction Arrangements design

Two main steps have been taken toward structuring and designing IIAs as presented in Figure 8. They are two reasoning rounds being grounded on the results of the literature review, where the three behaviors (Identification, Sensing, and Actuation) have been input to the first step and the 29 IoT properties (e.g., Adaptability, Context Awareness, Autonomy) to the second one. The next sub-sections detail the execution of this process, presenting the resulting IIAs from each step as well as application examples to illustrate the instantiation of each arrangement.

4.2.1 Reasoning on the three IoT Behaviors

In this first step, is has been given a focus on each of the three behaviors with the following goal:

a) Identifying logical relations among things playing each one of the three behaviors;
b) Identifying relations with other non-IoT elements (e.g., software systems that could operate or external actors that could intervene in the system).

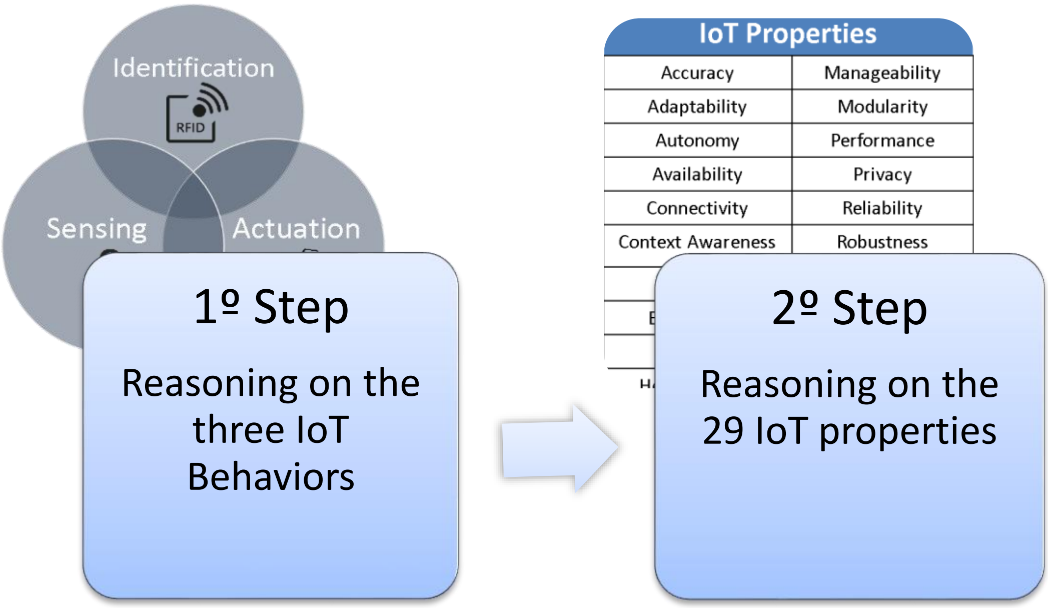


Figure 8 - Process toward structuring and designing IIAs

In other to assist the stipulated goal, things have been organized into Data Producers and Actuators according to the behaviors they play. These two fundamental elements guided the reasoning in this first step, which is detailed in the next sub-sections.

4.2.1.1 Data Producers

"Things" performing the **Identification** behavior can be considered as data producers given the reading and sharing of tags information. Similarly, "things" performing the **Sensing** behavior are also data producers once they sense and collect information from the environment. These have been put together in a higher-level element nominated **Data Producer**. It means that, in the moment of arranging the IoT interactions, things performing behaviors of Identification and Sensing are being treated as data producers equivalently, mainly because they play the same higher-level task and thus interact similarly with other components.

This organization contributed to simplifying the reasoning about possible interactions, leading to an understanding that Data Producers are supposed to share their findings with data consumer elements. These are presumed to be non-IoT ones as they do not perform any of the three behaviors. Instead, they only consume data for specific purposes, data exhibition, or decision-making, which are discussed below.

Those elements that consume data for exhibition purposes have been called **Data Exhibitors**. It means devices that enable data visualization by individuals who are candidates to interact with Data Producers. Therefore, considering the interaction among these two elements - Data Producer and Data Exhibitor – with the purpose of "Data Exhibition," the first IoT Interaction Arrangement has been designed, the so-called IIA-1[7] (Figure 9). This first and most basic interaction arrangement relies on data collection from Data Producers (sensors or tag readers) where data is made available to be visualized by users from their devices' displays (these devices have been referred to as "Human interface devices" (Hid) throughout the text).

Applications that instantiate this arrangement – especially by gathering data from radio frequency tag reading – exist since the term IoT has been coined by the Auto-ID Labs[8] together with the EPCglobal[9]. A supply chain management software was developed by Auto-ID Labs researchers that realized the importance of considering real-time information visibility as one of the crucial factors for the efficient supply chain management. The primary objective of the software on that occasion was to enable

[7] The designed IIAs have been numbered in order to facilitate mentioning them in the course of the text.

[8] Auto-ID Labs: https://autoidlabs.org/ (visited in August 2018)

[9] EPCglobal: https://www.gs1.org/epcglobal (visited in August 2018)

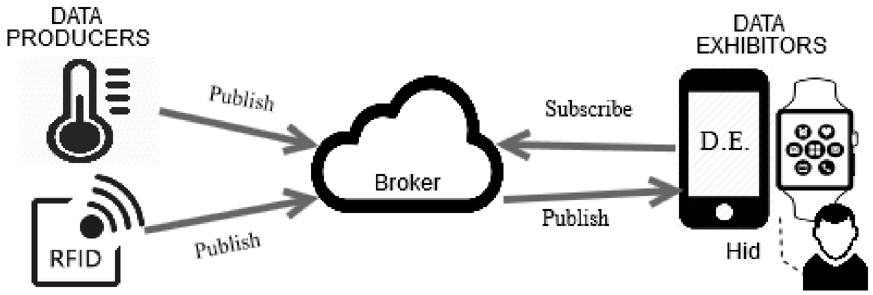


Figure 9 - IIA-1: IoT Data Exhibition

object's tracking in the supply chain for companies, by reading and sharing data from object's tags so that they can be read/visualized from interface devices [44]. This software can be considered an actual instantiation example of the IIA-1, as well as systems of the previously described scenarios for temperature and heartbeat monitoring.

The interaction among elements in all arrangements has been designed considering a Publish/Subscribe architecture, in which data exchange depends on a Broker containing "topics" by which publishers can send information at a specific topic and subscribers receive automatic messages in a subscribed topic [45]. For elucidation purposes, it is not in the scope of this work to discuss IoT architectures. Publish/Subscribe architecture has been chosen to represent interactions in the arrangement design only for an illustration matter, as it has been suggested as one architecture that meets IoT requirements better than other ones such as Request/Response architecture [45], treating, e.g., decoupling properties enabling interacting parties having no knowledge of each other [64].

4.2.1.2 Actuators

Related to "things" playing the **Actuation** behavior. By following this primary rationale, Actuators can receive commands from human actors by HIDs, leading to the design of IIA-2 in which actuation is triggered by individuals (Figure 10). It refers to functional units or systems where individuals trigger actuators' actions whenever they want to, not relying on data collected but just on actors' wills. For instance, a system for

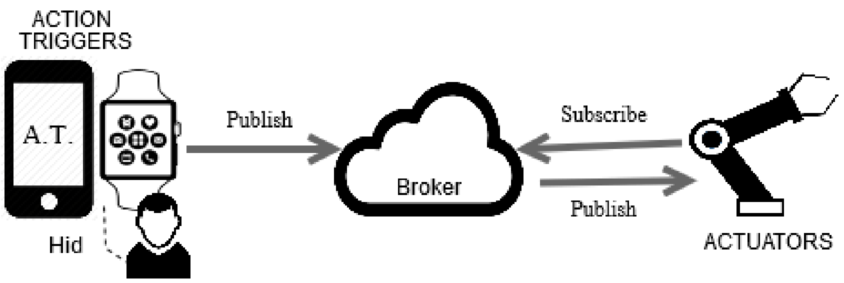


Figure 10 - IIA-2: Actuation triggered by an individual

operating light lamps in which it is possible to turn the lamp on and off (from an actuator) when the user requests it somehow from his/her tablet. Another example is the case in which people can use a mobile phone to raise and lower projection screens in conference rooms in an IoT-enabled hotel and conference center [92].

The rationale in which humans can trigger actuators' actions has led to another one where not just humans can trigger actions, but it can also occur using software systems commands. Therefore, software system elements can interact with actuators, composing the IIA-3 (Figure 11). This arrangement represents situations in which software systems trigger actuators' actions instead of individuals (humans). Considering the previous application example for turning lamps on and off, the actor is a software system, which can turn on and off the lamp programmatically.

Another rationale emerged from IIA-1 and IIA-2, with the possibility of having IoT components on both sides. Considering the first one - where individuals visualize data collected from IoT data producers (sensing and tag reading) - individuals can make decisions of triggering systems' actions based on the visualized/analyzed data. Similar to IIA-2, the system's action may be the one from IoT actuators. Thus, this flow of visualizing IoT data, making decisions, and manually triggering actuator's actions have led to the design of IIA-4 (Figure 12). This interaction arrangement represents functional units where IoT data support individuals on decisions of triggering actions. A system to support *urinary bladder* functions [22] can be assumed to exemplify instantiation of IIA-4. It is a medical application composed of three major components, a sensor, a display (HID) and a motor (actuator) in order to tackle neurogenic bladder. The flow starts with data collection to monitor the patient's urine level from a sensor; data is exhibited to the patient in treatment considering the bladder's filling level of 50%, 75% and 80% full; the patient visualizes the bladder levels at the display and are charged with taking decision to trigger the motor (actuator) for the emptying of the bladder.

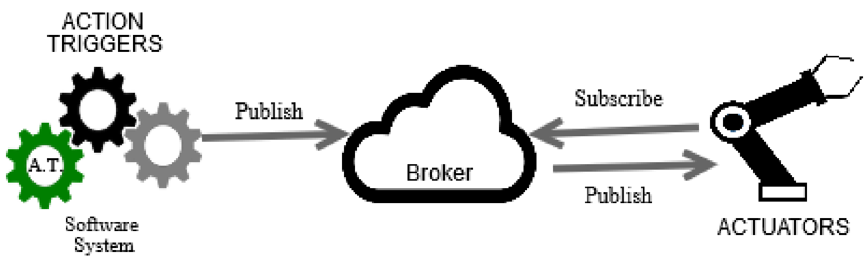


Figure 11 - IIA-3: Actuation triggered by a software system

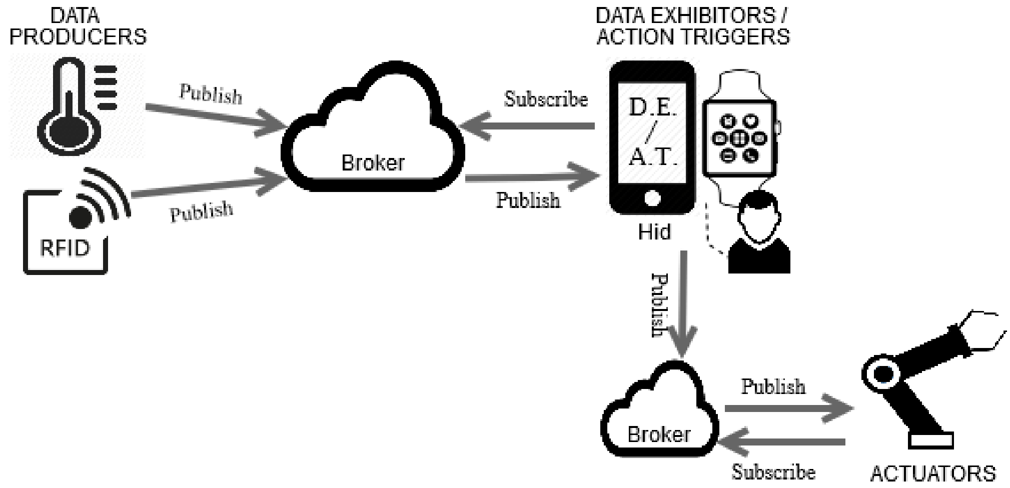


Figure 12 - IIA-4: Actuation triggered by an individual, based on IoT data.

These four IoT Interaction Arrangements could be designed from this first reasoning round based on the three IoT behaviors. In the sequence, the reasoning round based on the IoT properties is presented, which has also contributed to discovering IoT arrangements.

4.2.2 Reasoning on the 29 IoT properties

One concern in searching for IoT properties in the literature review relies on the assumption that the IoT properties could support the emergence of issues to be considered when constructing or tailoring models, tools, methods or practices – for requirements engineering, development, testing or deployment – so that they are feasible for the IoT development context. Each IoT property has passed for reasoning in order to capture **functional traits** intrinsic in them by which we could thus identify relations among IoT and non-IoT elements, and resulting interaction flows considering it.

It is meant by **functional trait** any clue/smell/peculiarity, from the essence of each IoT property, which has the potential to influence in the manner the system or a functional unit behave. For instance, considering the “Addressability” property, we focused on how it can influence systems in a functional sense. It resulted in an argument that Addressability concerns a network connection issue which does not impact the behavior of objects. As it was not possible to notice functional traits that could lead to IoT or non-IoT elements interacting in a flow, then Addressability will not be considered as a contribution to reach the goal of this step.

The same procedure described above has been performed for each IoT property. Although this list of 29 found properties contains vague definitions or terms that are subject to multiple interpretations, there was an effort to perform a reasoning procedure by focusing on the meanings discussed in the sources from which those properties have been extracted. For instance, “Efficiency” is open to many interpretations, being related to an object’s data collection efficiency, energy-efficiency, security-efficiency, information processing efficiency as well as service adaptability-efficiency. There are also properties whose sources have not presented clear concepts, and in this case, a focus has been given on the concepts and perspectives that are commonly addressed from the technical literature. This reasoning round is detailed as follows: for some properties, comments have been reported on their meanings, followed by reasoning and a resulting argument indicating the functional trait identified (or not), and the resulting IIAs. The properties were organized according to objects or systems relation and ordered according to their contribution in designing interaction arrangements.

4.2.2.1 Objects’ properties

- Addressability and Unique ID: In general, these properties concern to the capacity of one entity to be targeted and found once associated with an identifier not associated with any other entity in the system.
    - Reasoning: Even though Addressability and Unique ID are usually related to Network Connectivity [7,8,14,36,51,57], it is possible to reason abstractly and look at those properties from either a “Virtual” or “Physical” lens. The Virtual one is about identification for connectivity purposes in which objects need to be uniquely identified within a network from IP addresses, URLs, URNs, etc. On the other hand, from the Physical lens, not fixed objects are identified by tags to be read in strategic places or from readers spread along with the physical environment. This sort of identification brings the advantage of physical/material traceability. The object’s tag reading enables retrieving physical location and the status of a tag holder (its temperature, rotation, vibrations, etc.) or the environment in that right location (pressure, noise, humidity, and others). This physical identification viewpoint of addressability can give us a functional trait related to the need of collecting tag ’s data and sharing it for some purpose.
    - Functional trait: Need for collecting tag’s data and sharing it for some purpose. This functional trait leads to the “Data Exhibition” arrangement already designed in the last step focused on the Identification behavior itself.
- Autonomy: Autonomous objects may be interpreted as those capable of performing many functions (roles), e.g., the perception of the environment, self-configuration, actuation, and many others. For example, there are sensors that not only sense the

environment but also are equipped with autonomous and proactive behavior skills, context awareness, collaborative communication, etc. [7].

  - Reasoning: The fact of coupling several capacities in a single device is assumed to be a technical project decision, not exactly a concern of functional analysis. Also, since objects are autonomous (at different levels), they may not necessarily depend on direct intervention from users, but the overall system may do (discussed in the system's property).
  - Functional trait: it was not possible to identify.

- Mobility: According to [14], a large part of IoT objects are not location-fixed, but have a certain degree of mobility.
  - Reasoning: This characteristic raises challenges related to connectivity and how keeping intermittent network connections; that is, it leads to the need for caring about objects' visibility in the network. Mobility does affect the way the connection/interaction is established and kept, but it does not affect the need for things playing specific behaviors interacting inflow, being not possible to identify functional traits from this property.
  - Functional trait: it was not possible to identify.
- Smartness: This is a general characteristic since diverse abilities may turn an object smart at different levels. When a "smart object" is mentioned in the literature, it is often associated with the ability to be autonomous and adaptable, minimizing human intervention. In the context of Social Internet of Things (SIoT), "smartness" could also mean "objects that have a social consciousness and exhibit social behaviors allowing them to build their social network of objects [14]". A known term commonly used to refer to a smart object is *spime*.
  - Reasoning: Because "smartness" is too abstract, we cannot argue which things are supposed to operate in order to realize this property, or even whether there will be a human intervention (or not) or some specificity in some system's behavior when composed of smart objects.
  - Functional trait: it was not possible to identify.
- Visibility: According to Atzori *et al.* [7], visibility stands for the traceability of objects and the awareness of their status, current location, etc.
  - Reasoning: Just as the rationale for Addressability when observed from a "Physical" lens. It is necessary to tag this object or to sense it somehow to turn an object visible. The data collected needs to be shared for the awareness of interested ones. This flow itself is the functional trait observed when reasoning about Visibility.

- Functional trait: Sensing or Tag identification of objects in order to make available their status. It has already been thought in the first step resulting in the “Data Exhibition” arrangement.

4.2.2.2 System’s properties

- Context Awareness: Abowd *et al.* [25] have defined Context-Awareness as “the use of context to provide task-relevant information and/or services to a user.” They also have defined context as “any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or a physical or computational object.” That is, context-aware systems capture context information and reacts autonomously according to collected data. For data collection activity, there are different sorts of information sources that context-aware applications may base on: (1) explicit information provided by users, (2) those stored in contextual knowledge bases, (3) information inferred through reasoning, and also (4) those perceived from the environment [85]. Also, reactions can be of the following types: (5) adaptation or variation in the behavior of the system, responding to changes in the environment and the actions/definitions of users (e.g., personalization of interfaces and content); (6) assistance in executing the task being performed, e.g., alerting users about actions they should perform to achieve their goals; (7) notifications about context perception, which refers to notifying users about the context associated with people and interactions of their interest, related to the executing task, supporting them to coordinate their own actions; and (8) other services, such as the use of the context to semantically enrich the knowledge managed by the application.
    - Reasoning: From this concept, these three activities are crucial for context-aware systems: data collection, decision-making, and reaction. Thus, context-aware systems based on IoT can carry either Identification, Sensing or Acting behaviors. It can be argued that context-aware IoT applications that use information perceived from the environment (4) are those who carry sensing or identification behavior, that is, there should be sensors or tag readers to capture information from the monitored environment, while data collection from the modes (1) (2) or (3) can be performed by non-IoT elements. In both cases, there is a need for elements with the responsibility of taking automated decisions on data and triggering system reactions based on contextual information. Concerning the system reaction, IoT actuators are supposed to have their role performed in this activity as a consequence of the collected context information followed by decision-making.

- Functional trait: Capturing context information and reacting according to information collected. This functional trait involving the Context-Awareness property has led to the identification of interaction flows among elements with the following responsibilities:
    - Data collection: considering those mentioned types of information sources, data producers can be IoT elements (sensors and tag readers that perceive the environment) or non-IoT ones (software systems in general responsible for the other three types of information collection (1) (2) or (3)).
    - Decision Making and Action triggering: non-IoT elements, i.e., software systems that take an automatic decision on data and trigger actions. Differing from IIA-4, in context-aware systems the decision-making and action triggering tasks are automatic, needing software-based components to perform them.
    - Actuation Performing: IoT elements (actuators) performing system reactions related to the (5) type; or non-IoT (software systems) with the responsibility of performing other actions (6) (7) or (8).

Before presenting the resulting IIAs, it is essential to highlight that systems can be characterized as IoT-based when IoT elements operate in either data production or actuation sides. This statement follows the IoT concept presented in the previous chapter:

> *"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."*

Hence, the other five IIAs could be designed. They are explained as follows, and application examples are given.

Following the sequence, the IIA-5 represents functional units for "Actuation triggered by software systems based on IoT data" (Figure 13) and brings together IoT elements in both sides. That is, sensors or tag readers are data producers; data is shared with a software system that takes decision and triggers actions; the flow finishes with actuators performing their specific actions. This arrangement, as well as the subsequent ones, clearly illustrates the functional trait that involves context-aware systems. In this case, IoT elements capture context information, software systems trigger actions based

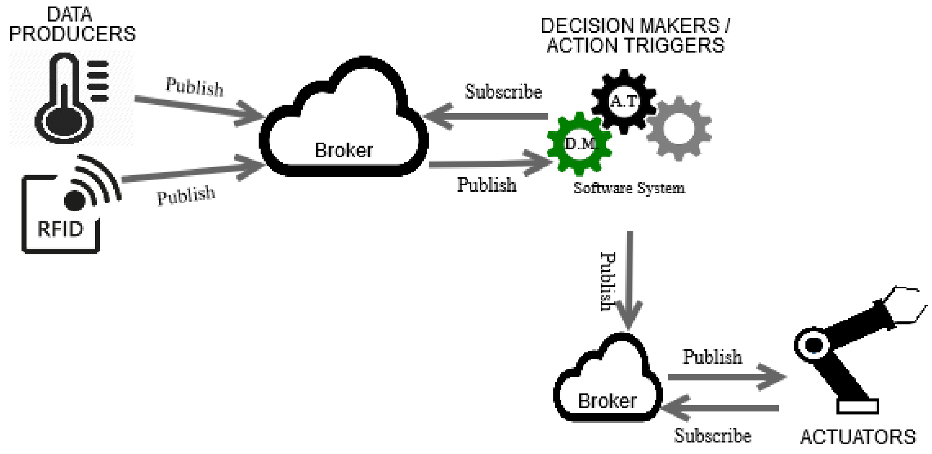


Figure 13 - IIA-5: Actuation triggered by a software system, based on IoT data.

on decisions, and the system reacts according to information collected. Kolokotsa *et al.* [47] present an intelligent environment and energy management system for greenhouses, which is a relevant one to illustrate the instantiation of IIA-5. The system monitors greenhouse's indoor luminance, temperature, relative humidity, CO^2 concentration, and the outside temperature; the system analyses and takes a decision on data based on fuzzy logic, and actions are triggered to control heating units, motor-controlled windows, motor-controlled shading curtains, artificial lighting, CO^2 enrichment bottles and water fogging valves.

Context-aware systems are not only based on IoT data as mentioned before. Software system elements may perform data collection activity as well. In such cases, the actuation should be performed by IoT elements so that the system can be characterized as IoT-based. From this statement, another arrangement has been designed, the IIA-6 (Figure 14) for actuation triggered by a software system based on non-IoT data. An assumed system for automated doors locking of data centers can be used to exemplify this arrangement. Data producers are software systems that collect data of server authentication; the doors are locked (actuation) after the decision-making software identifying three failed authentication attempts to the server.

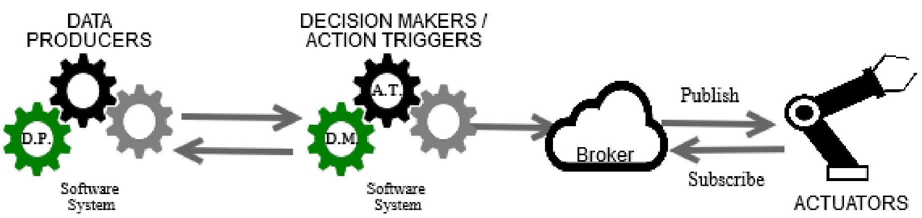


Figure 14 - IIA-6: Actuation triggered by a software system, based on non-IoT data.

There is also the possibility of existing non-IoT elements performing the system reaction activity. In cases where IoT data collectors are part of the system, the action based on decision-making may not necessarily be performed by IoT actuators, as shown by IIA-7 (Figure 15). The instantiation example of this arrangement can be the Twitter-enabled sensing system called *Botanicalls* [28]. It aims to keep individuals informed about the health state of their plants. The flow starts with the collection of moisture information from a sensor (IoT data producer) that goes into the soil for measurement; data is shared with a software system with the responsibility of making some decision based on data and then triggering non-IoT action; the non-IoT action is the twittering of moisture information directly to the plant's twitter account, which is also performed by a software system (action performer).

Another example can be a system for real-time medical inventory. In a pharmacy of a hospital, the equipment and products (medicines, procedure gloves packs, mask packs, etc.) are automatically and real-time identified once they are attached with RFID tags. When some of those products are taken out of the pharmacy, the system makes the stock removal in the database.

The arrangements IIA-5, 6 and 7 are based on the context-aware property. It means that context-aware systems based on IoT are supposed to be (or one of their functional units) instances of those interaction arrangements. These results can be regarded as pictures of systems developed to contribute to the fully factories' automation, eliminating human intervention and improving performance. In details, they cover vital features such as sensing of industrial machines and products and advanced analytical systems to perform calculations, decisions and deliver insights on systems' reactions

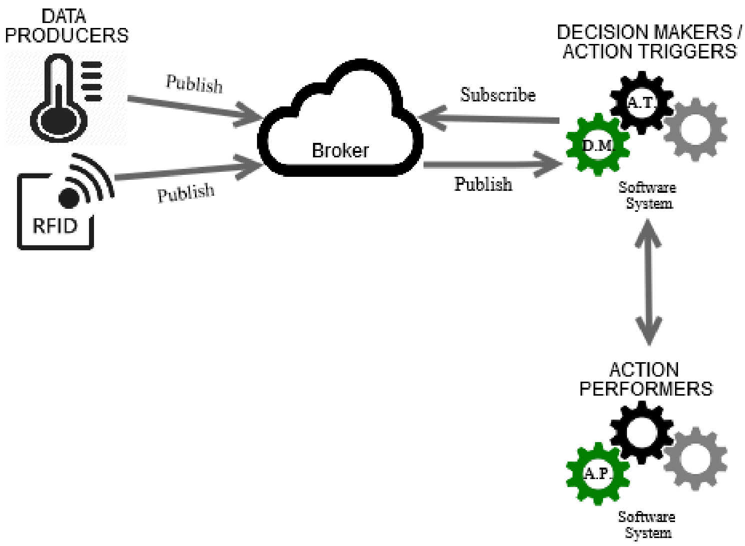


Figure 15 - IIA-7: Non-IoT actuation triggered by a software system, based on IoT data.

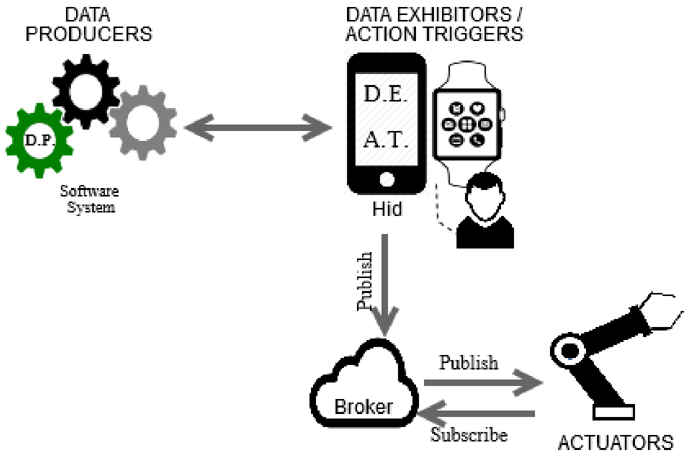


Figure 16 - IIA-8: Actuation triggered by an individual, based on non-IoT data.

and optimization, both strong characteristics of the Industry 4.0 era. According to Trappey *et al.* [82], IoT is a core enabling technology that enables industries to move from the 3rd (beginning of robotics manufacturing) to the 4th era (beginning of Cyber-Physical Systems leading to advanced manufacturing). The IIAs 5 to 7 are important ones to be implemented for systems of the Industry 4.0 era as they can contribute to eliminating the need for human intervention, cutting the cost of operations over time.

The context-aware functional trait in conjunction with the IIA-4 has inspired and guided us to the design of two other arrangements having decision-makers/action triggers as a distinct element. Context-aware systems perform those tasks fundamentally by automatic means, but there may be systems where individuals take decisions on non-IoT data and trigger IoT actions, represented by the IIA-8 (Figure 16). Also, the opposite can also occur, wherein one side IoT data is collected, individuals take decisions of triggering non-IoT actions based on data visualized on their HIDs, IIA-9 (Figure 17).

A data center system can also be assumed to instantiate IIA-8. The difference is that data from failed authentication attempts can be shared with individuals in charge of infrastructure issues, is responsible for deciding locking or not data center's doors. A remote server control system can be assumed for remote server' shutdown (by prompt command) triggered by individuals once they visualize data of electric current sensors from their HIDs to illustrate the IIA-9.

- Autonomy:
    - Reasoning: In this case, autonomous systems may not necessarily depend on human intervention. Therefore, interaction structures with no human intervention might be chosen.
    - Functional trait: functionalities demanding no (or low) human intervention.
- Accuracy: This property is commonly related to data or operation quality.

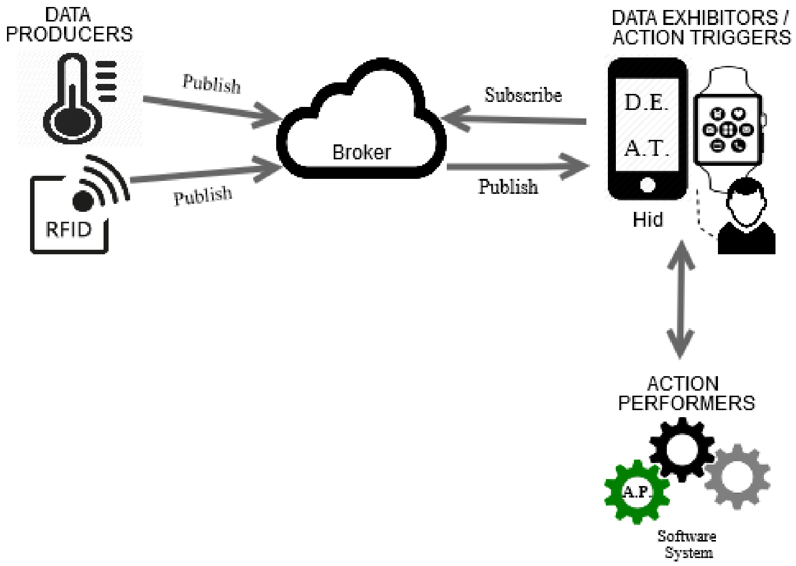


Figure 17 - IIA-9: Non-IoT actuation triggered by an individual, based on IoT data

- Reasoning: The desired level of data or operation accuracy does not influence the system considering the three IoT behaviors, but the system quality instead.
- Functional trait: it was not possible to identify.

- Flexibility: It is related to systems operating in open scenarios, "in which new functions/capabilities/services need to be accommodated at run-time without having been necessarily captured at design time" [57]. There may be "open scenarios" due to technology volatility (technologies emerging every time), business objectives, threat and non-fixed operational context in general. Self-configuration, self-organization, self-optimizing, and other characteristics are both enablers of Flexibility, and all of them are thresholds to achieve another property: Ubiquity.
  - Reasoning: To the best of our knowledge, even though one system is capable of operating according to some runtime status, it must exist a space of system functions which are like "juggling pieces" handled by the system's adaptation management mechanism. These functions are meant to be invoked and controlled in response to issues of different natures: networking, energy consumption, privacy, and business-related issues. Users running location-sharing apps need services to be flexible enough to manage instabilities of wireless connections so that they do not get their tracked object/person lost to give an instance of networking nature issues. From the business nature, sleepy drivers require their "smart" cars to be flexible in the sense of pondering the triggering of abrupt or gradual breaking functions by taking into

account objects' distance, velocity, wet or dry tracks. However, similar to the Smartness property, it is not possible to identify which IoT behaviors a told "flexible" system may have.
    - Functional trait: it was not possible to identify.
- Adaptability:
    - Reasoning: This property can mean the ability to adapt in response to runtime-system exceptions or the sort of adaptation related to context changes. It is hand-to-hand with flexibility.
    - Functional trait: it was not possible to identify.
- Availability:
    - Reasoning: It may be in the sense of high availability of systems, or the availability in the sense of access permission. However, it does not influence the system's IoT behavior.
    - Functional trait: it was not possible to identify.
- Connectivity: The myriad things should be addressable, accessible, and controllable, and the connectivity should ensure that.
    - Reasoning: As it is primarily an operational property, it does not need IoT behaviors for its realization.
    - Functional trait: it was not possible to identify.
- Efficiency:
    - Reasoning: As discussed before, even though this property can lead to many interpretations (energy efficiency, data analysis efficiency, etc.), it is primarily operational, being not possible to identify the presence of IoT behaviors.
    - Functional trait: it was not possible to identify.
- Extensibility:
    - Reasoning: The capacity of a system to be extensible, making possible adding new components and ensure at the same time the interoperability, performance, connectivity, etc. Those components to be added are supposed to be sensors, actuators, etc. Adding new components when a system is already deployed may require the updating of requirements artifacts. However, extensibility does not bring issues in the functional requirements specified up-front (design time), including the definition of IoT behaviors.
    - Functional trait: it was not possible to identify.
- Heterogeneity:
    - Reasoning: it is primarily related to various technical properties among objects, that is, different protocols, connectivity, data format, energy consuming, etc.

    - Functional trait: it was not possible to identify.
- Interoperability: The work of Motta *et al.* [60] proposes a rethinking of Interoperability concepts against a global digitalization era with different devices engaging new interactions and composing Systems of Systems (which is consonant with IoT). From this point of view, they stated Interoperability is “the ability of things to interact for a specific purpose, once their differences have been overcome.”
    - Reasoning: Even considering an Interoperability concept consistent with the current digitalization era, this is mainly an operational property in which no functional traits related to the IoT behaviors could be noticed.
    - Functional trait: it was not possible to identify.
- Manageability: Heterogeneity brings challenges for Manageability, having to manage Fault, Configuration, Accounting, Performance, and Security [1].
    - Reasoning: This property is mainly operational, then it was not possible to identify IoT behaviors operating it.
    - Functional trait: it was not possible to identify.
- Modularity:
    - Reasoning: This characteristic goes hand to hand with Interoperability. It is mostly related to the hardware/physical layer and architecture. In one hand, Modularity means the ability to build a smart object (or an Internet connected object) by putting different modules (e.g., sensors or actuators) produced by different manufacturing companies together without getting restricted to one vendor. On the other hand, it allows organizations to focus on one component of the IoT architecture and become experts on that, rather than having to build end to end solutions, besides providing more choices and options for consumers as to which modules to choose based on specific constraints [14]. From the reasoning on this property, it was not possible to identify the presence of IoT behaviors.
    - Functional trait: it was not possible to identify.
- Performance, Privacy, Reliability, Robustness, Scalability, Security, Sustainability, Traceability, and Trust:
    - Reasoning: These properties are mainly operational, being not possible to identify the need for IoT behaviors to realize them.
    - Functional trait: it was not possible to identify.

4.3 Chapter Considerations

From an initial conjecture on the relevance of raising a set of interaction patterns to assist on the investigation of this work, a process of reasoning has been carried out in order to find common elements and recurrent interactions among them to design the interaction patterns that are called IoT Interaction Arrangements, the IIAs. This reasoning process has been grounded on the three IoT behaviors (first step) and the 29 IoT properties found in the literature review (second step).

Four IIAs have been designed as a result of the first reasoning round based mainly on two higher-level IoT elements: Data Producers and Actuators. From the second round other five IIAs have appeared, having the Context-Aware property as the most contributor. Summing up the findings, nine domain independent IIAs have been designed, as listed below:

- IIA-1: IoT Data Exhibition
- IIA-2: Actuation triggered by an individual
- IIA-3: Actuation triggered by a software system
- IIA-4: Actuation triggered by an individual, based on IoT data.
- IIA-5: Actuation triggered by a software system, based on IoT data.
- IIA-6: Actuation triggered by a software system, based on non-IoT data.
- IIA-7: Non-IoT actuation triggered by a software system, based on IoT data.
- IIA-8: Actuation triggered by an individual, based on non-IoT data.
- IIA-9: Non-IoT actuation triggered by an individual, based on IoT data.

These IIAs are high-level representations of the interaction among IoT and non-IoT components in order to accomplish some goal, such as data exhibition in HIDs, actuation manually triggered by an individual, and others. As discussed in the introduction of this chapter, the content of IIAs (elements and flows) may assist in the scenario description activity in IoT-based projects. From them, analysts can have a vision of relevant information to be captured in the scenario descriptions based on IIAs' content.

Relevant information from IIAs has been identified and brought together in one document related to each IIA. These documents are catalogs of suggested information to be captured in scenario description activities. The subsequent chapter presents these Catalogs, the relevant information from IIAs and Guides to help analysts finding the best-related arrangement for scenarios to be described.

5 Toward an IoT scenario description approach

This chapter aims at presenting the approach for IoT scenario specification grounded on the IIAs described in chapter 4. We understand that the interaction clarification among parts of IoT-bSS brought by each IIA can reflect the scenario's core in its definition: a temporal sequence of actions. Further, specifically focusing on each IIAs element, these elements are potential sources of information to capture in scenario descriptions, bringing the IoT realm properties to scenario specification. Catalogs compile all of this information, which composes the proposed ***ScenarI$_{oT}$*** *software technology.*

5.1 Introduction

Once a more in-depth understanding of IoT realm has been obtained from the literature review (Chapter 3), IIAs could emerge by observing current flows among IoT elements. IIAs composition raised when reasoning about the IoT concept, properties and a set of three behaviors, following the understanding that things play finite behaviors of identification, sensing or acting despite the large amount and diversity of devices.

The set of raised IIAs supports conjecturing that, in general, scenario descriptions represent instantiations of one IIA or a combination of specific ones. That is, **scenario descriptions could present information of temporal action sequences and the involved elements respectively based on (a) the explicit IIAs` interaction flow and (b) the interacting elements (data producers, individuals, actuators, among others)**. As the flow of interaction is naturally exposed in the IIAs, what remains is to work on (b), that is, extract information intrinsically related to the IIAs' elements and capture it in the scenario descriptions, bringing thus the IoT realm to the specification of software systems.

Before performing the process of extracting information from IIA's elements, it is crucial to stipulate the scenario approach boundaries according to behaviors objectives. That is, before developing this approach, it is essential to classify/characterize it taking into consideration the expected behaviors and possible configurations. As mentioned before, scenarios have no defined content if they have no purpose, as *scenarios' content can be composed of several types of information on different abstraction levels* [53]. Thus, a scenario approach classification is presented in the next subsection, followed by subsection 5.3 describing the information extraction from the IIAs' elements and the information compilation into Catalogs. This chapter finalizes in subsection 5.4, which presents the whole scenario description support named **ScenarI$_{oT}$.**

5.2 Scenario description approach classification

The **ScenarI$_{oT}$** approach employs the 4-dimensional framework from Rolland [71] as the classification basis. As explained in Chapter 3 and depicted in Figure 2, scenarios can be classified considering four views: **form**, **purpose**, **lifecycle,** and **content**. A question resumes each view, and each question was brought to the **ScenarI$_{oT}$** approach context in order to guide its classification as follows.

- Why using a scenario of the **ScenarI$_{oT}$** approach?

This question from the **purpose view** has its answer on one of this research's objectives: to support IoT-bSS specification by capturing IoT properties in scenario descriptions. Therefore, the **ScenarI$_{oT}$** aims at being a complement to specify and communicate systems behaviors using scenarios. The utility of scenarios *per se* for these matters has been presented in Chapter 2, but in summary, it is because scenarios "can be a remedy for the complexity and/or a complement to assist understanding and documenting systems' behaviors" [3]; also they are a "natural means for writing partial specifications" [33]. However, it can also be used in later phases according to the coherence among the phase purpose and the properties of the artifacts, but this matter is not in the research scope of this work.

- How to manipulate a scenario of the **ScenarI$_{oT}$** approach?

Considering the **lifecycle view** and attempting to maintain consistency with the purpose stated above, the scenario description artifacts are supposed to be employed in early-phases of IoT software systems projects. Then, artifacts start to materialize (created) in the early phases of projects but are not limited to it. Scenarios can be reused in later phases having, therefore, a potential to have a *persistent lifespan*, but it is an open issue. That is, the artifacts can be created from scratch with the support of the **ScenarI$_{oT}$** approach, and can evolve throughout their lifecycle to keep themselves aligned with the purposes and goals of their primary roles. However, the artifacts can also represent *transient scenarios* when used only in a specific phase, being disposed of after their usage.

- In which form is a scenario of the **ScenarI$_{oT}$** approach expressed?

A scenario assumes a **narrative form** in this work considering one of the motivations for the research problem, which is aligned with the intents stated in the purpose view: communicating systems' behaviors so that stakeholders from different areas and knowledge levels could understand and contribute to discuss the requirements. This choice is also explained due to indications found in the technical literature such as: “narrative texts are considered convenient for both writing and reading, easy to understand for domain experts, analysts and systems developers” [12]; also, “stakeholders do not need to learn formal syntax to describe and understand them [scenarios]” [55].

- What is the knowledge expressed in a scenario of the **ScenarI$_{oT}$** approach?

This question represents the core of our research problem. Our research objectives are directly and mainly related to the **content view** specifically. Scenarios produced by the **ScenarI$_{oT}$** approach are supposed to store information about IoT-bSS behaviors so that they can be useful in software projects with IoT realm. On the *abstraction facet* of scenario contents view, our approach intends to concentrate on details, that is, to produce *scenarios instances* grounded on an IIA or a combination of them. Following this reasoning, the IIAs presented in the previous chapter link to or regard as *Type of Scenarios* because they describe interaction flows and the elements with little concreteness. That is explained because the IIAs use categories such as "Data Producers," "Action Performers" to express the elements with no actual names. The goal of the **ScenarI$_{oT}$** approach supports the description of scenarios with accurate information.

Among the remaining facets, the *context*, *argumentation* and *coverage*, the last one is eligible to be discussed in this classification because it is directly related to the purpose view configuration. That is, considering that the primary purpose is specifying and communicating systems' behaviors employing scenarios so that stakeholders can engage in the process, then the coverage facet focuses on behavioral aspects rather than structural for instance. The context and argumentation facets are both broad issues and could not be considered in this work’s scope, in particular the context facet, which has been studied due to e.g., nebulous tasks of stating what context information should be considered to provide task-relevant information and/or services to a user, following the context definition from Abowd *et al*. [25].

To summarize the **ScenarI$_{oT}$** approach classification, it aims at complementing IoT-bSS specification; the artifacts born in early project phases and can evolve or not, depending on their usage in later activities; scenarios are expressed under a narrative form and have their containers filled with IoT properties, capturing interaction among elements and further information extracted from them.

This classification will be the baseline to extract information from the IIAs in order to compose our scenario description approach. This extraction is presented in the following subsection.

5.3 Catalogs capturing IoT Information

The process of extracting information from each element present in the IIAs was carried out in an ad-hoc way by questioning which information would be relevant to be captured in the IoT scenario descriptions. A piece of relevant information is concerned with the statement that scenarios are intended to be written in early IoT project activities, needing to be accessible and understandable for stakeholders from non-technical areas.

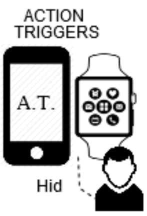
As explained previously, each IIA can hold a *type* of *scenario*, and thus can ground a top-down scenario instance writing. Type scenarios describe abstract entities as is the case of the entities in the IIAs, which nominate them from their roles or categories. In light of this, each abstract entity was examined in order to extract inquiries that can support raising concrete (or less abstract) information to compose the scenarios artifacts. The results are listed in Table 5 and Table 6. For a matter of organization, the first table was restricted to information of IoT entities, whereas Table 6 describes information from the other ones. In both tables, each entity is represented in the “Abstract entity” column, followed by a “Discussion” column reserved for an explanation about the concerned entity and the inquiries extraction procedure. Finally, the resulting inquiries are listed in the “Related information” column.

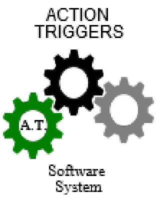
Table 5 - Extraction of IoT entities' related information

IoT Abstract Entity	Discussion	Related Information
	This abstract entity (explained in Chapter 4) represents those things with **sensing** and **identification** behaviors. 1) The first relevant information to be concretized is who collects data from the universe composed of sensors and tag readers; 2) Once revealed the type of data producer, it can be interesting to define the type of data that is going to be collected, e.g., temperature, humidity, among others 3) Another information that can be captured is the source of data. For instance, taking temperature data as an example, it can be collected from the environment (e.g., rooms, parks, cities) or from specific objects (e.g., a coffee mug, refrigerator, among others)	1) Who collects data {Sensors, Tag readers} 2) What data is collected 3) Source of data
	1) Following the reasoning of the previous discussion, the "actuators" abstract entity could also be concretized, but in this case, only actuators compose the universe of things. 2) Actuators can perform several mechanical interventions in the real world. Therefore, it can be interesting concretizing the type of action (e.g., circular motion, straight-line motion, On/Off circuit, among others).	1) What performs action {IoT actuator} 2) Type of action

Table 6 - Extraction of Non-IoT entities' related information

Non-IoT Abstract Entity	Discussion	Related Information
	In our interpretation, information can be extracted from this abstract entity taking into consideration the device (data exhibitor) and the individual consuming data. From the **data exhibitor perspective**, the following information can be captured: 1) The specific device expected to exhibit data that is a device running a user application,	**Data exhibitor:** 1) What exhibits data 2) Data format **Data consumer:** 3) Who accesses data 4) Data semantics

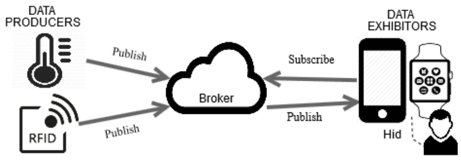
	e.g., smartwatches, smartphones, or even more futuristic devices such as electronic glasses. Realizing the actual device or platform where users are going to access data may be essential to assist requirements discovery and late project decisions. 2) The format in which data is intended to be displayed is also linked to the sort of display devices. The format may be interesting information to be captured because it may vary according to display limitations and other issues. From the **data consumer** perspective, the following information can be captured: 3) Which individual is going to consume data? It can be a person with an actual name, a persona, profile, or even a role. 4) The semantics of data may be explored regarding each's expertise, experience or interests. That is, the meaning of data is related to the perspective of those who are expected to access it. For instance, in a farm low-temperature values may be harmful to someone and positive for others. A veterinarian may associate low environment temperature values as the need to be alert on the health of newborn animals, whereas this low-temperature values may be positive information for people in charge of assuring animals' feed/food quality.	
	The inquiries of this non-IoT abstract entity are equivalent of IoT data producers. The difference is that IoT has well-defined behaviors. In this case, the universe of data collectors' types is open-ended. The data collector can be concretized as a software system, an information service, a computer and entities that collect data by conventional means. The main issue is that, in our interpretation, it may be confusing or useless defining a concrete entity such as a service name, program or even a system module name referring to this abstract entity in early phases of projects. Therefore, this entity might be referenced as less abstract.	1) Who collects data {Non-IoT data collection} 2) What data is collected 3) Source of data
	This entity may be considered similar to the "Data exhibitor" one. However, in this case, the focus is not on data itself and how it will be treated. The focus is on: 1) from which type of device/platform the action will be triggered (smartphone, TV, Smartwatch, among others)	1) What does interface with individuals? 2) Who triggers action

	2) the individual being the action trigger (person, role, among others).	
	An abstract entity holds two different roles with the responsibilities of making decisions and triggering actions. These roles can be treated separately in the design or development phases of a project. For instance, in object-oriented design, each responsibility could be implemented as a behavior (method) of some class. 1) As explained before, in our interpretation, it can be interesting capturing the entity that is intended to perform these roles in a less abstract form considering this information is intended to be used in early phases of projects. 2) The circumstances for triggering actions may also be valuable information, in which the action triggering criteria need to be specified. For instance, considering the decision will be made on IoT data, some action will be triggered if, and only if, the environment temperature value rises above 55ºC.	1) Who takes decision and triggers actions 2) Circumstances for triggering action
	It relates to an entity that triggers actions programmatically (as explained in the composition of the IIA-3 in Chapter 4). That is, unlike the entity described previously, this one triggers actions in a programmatic form, with no decision on data itself. Similarly, it can also be the case of : 1) Concretizing (or specifying less abstractly) the action trigger (e.g., agent, program, software system) 2) Stating the programmatic circumstance for triggering actions.	1) Who triggers action 2) Circumstances for triggering action
	This entity is the equivalent of the “Actuators” one. In this case, actions are not IoT related, but they are everyday actions such as calling a system functionality (send an email, post a tweet on Twitter, sending a document to the printer, among others).	1) What performs action 2) Type of action

After extracting the inquiries from each abstract entity present in the IIAs, they were organized according to each arrangement and compiled into Catalogs (subsequent subsections). These catalogs intend to support the writing of scenario descriptions with background on the IoT realm. The related information is thus suggestions to be captured in the early phases of the project.

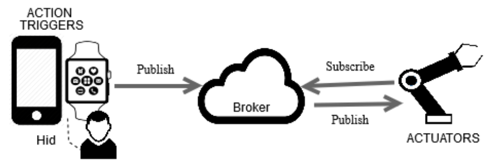
The whole approach is presented in the next subsection, where we explain the steps to write the scenario descriptions, from the identification of the candidate IIA to represent an abstraction of some scenario to how making use of the catalogs to support writing the artifacts.

5.3.1 Catalog for the - IIA-1: IoT Data Exhibition



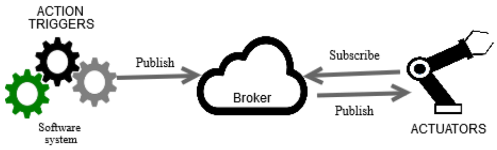
Catalog – **Data exhibition**	
Entity	Related Information
Data producers	1. Who collects data? {Sensors, Tag readers}
	2. What data is collected? (Temperature, humidity, among others)
	3. Source of data (rooms, coffee mug, refrigerator, ground, among others)
Data Exhibitors (Hid)	1. What exhibits data? (Ex. Devices running user applications)
	2. Data format
Data consumer (human)	1. Who accesses data? (Ex. A person, Persona, Profile, Role, among others).
	2. Data semantics (the meaning of data according to who visualizes it)

5.3.2 Catalog for the - IIA-2: Actuation triggered by an individual



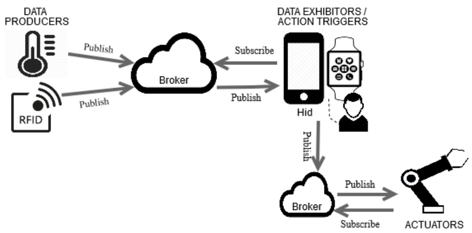
Catalog – Actuation triggered by an individual	
Entity	**Related Information**
Hid (display)	1. What does interface with individuals? (Ex. Smartphone, TV, SmartWatch, among others)
Action trigger (human)	1. Who triggers action (Ex. Person, Persona, Profile or Role)
Action Performers	1. What performs action {IoT actuator}
	2. Type of action (Ex. Circular motion, Straight-line motion, On/Off circuit, among others)

5.3.3 Catalog for the - IIA-3: Actuation triggered by a software system



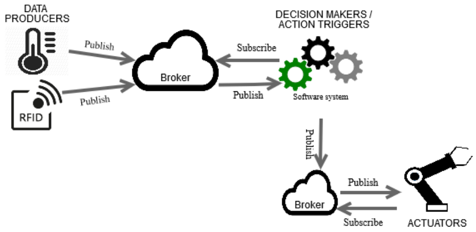
Catalog – Actuation triggered by a software system	
Entity	**Related Information**
Action Trigger	1. Who triggers action (Ex. Agent, software system) 2. Circumstances for triggering action
Action Performers	1. What performs action {IoT actuator}
	2. Type of action (Ex. Circular motion, Straight-line motion, On/Off circuit, among others)

5.3.4 Catalog for the - IIA-4: Actuation triggered by an individual, based on IoT data.



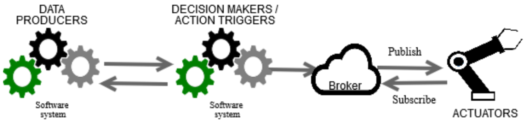
Catalog **–** Actuation triggered by an individual, based on IoT data	
Entity	Related Information
Data producers	1. Who collects data {Sensors, Tag readers}
	2. What data is collected (temperature, humidity, among others)
	3. Source of data (rooms, a cup of coffee, refrigerator, ground, among others)
Data Exhibitor (Hid)	1. What exhibits data (Ex. Devices running user applications)
	2. Data format
Data consumer and Action trigger (human)	1. Who accesses data (Ex. Person, Persona, Profile, Role, among others).
	2. Data semantics (the meaning of data according to who visualizes it)
Action Performers	1. What performs action {IoT actuator}
	2. Type of action (Ex. Circular motion, Straight-line motion, On/Off circuit, among others)

5.3.5 Catalog for the - IIA-5: Actuation triggered by a software system, based on IoT data.



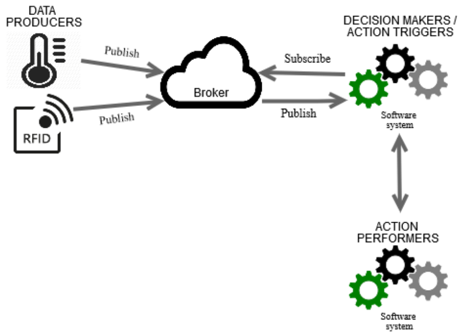
***Catalog* – Actuation triggered by a software system, based on IoT data**	
Entity	**Related Information**
Data producers	1. Who collects data {Sensors, Tag readers}
	2. What data is collected (temperature, humidity, among others)
	3. Source of data (rooms, a cup of coffee, refrigerator, ground, among others)
Decision maker / Action trigger	1. Who takes a decision (Ex. Agent, software system)
	2. Circumstances for triggering action
Action Performers	1. What performs action {IoT actuator}
	2. Type of action (Ex. Circular motion, Straight-line motion, On/Off circuit, among others)

5.3.6 Catalog for the - IIA-6: Actuation triggered by a software system, based on non-IoT data.



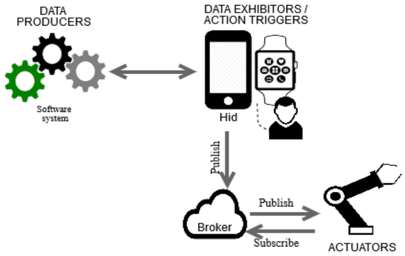
Catalog – Actuation triggered by a software system, based on non-IoT data	
Entity	Related Information
Data producers	1. Who collects data {Non-IoT data collection}
	2. What data is collected
	3. Source of data
Decision makers / Action trigger	1. Who takes a decision (Ex. Agent, software system)
	2. Circumstances for triggering action
Action Performers	1. What performs action {IoT actuator}
	2. Type of action (Ex. Circular motion, Straight-line motion, On/Off circuit, among others)

5.3.7 Catalog for the - IIA-7: Non-IoT actuation triggered by a software system, based on IoT data.



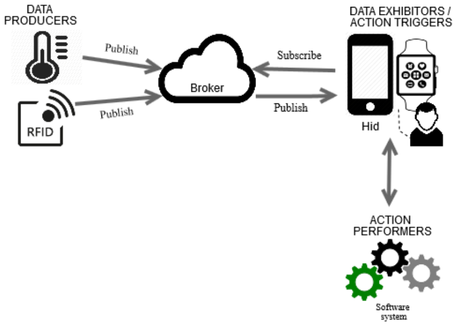
***Catalog* – Non-IoT actuation triggered by a software system, based on IoT data**	
Entity	Related Information
Data producers	1. Who collects data {Sensors, Tag readers}
	2. What data is collected (temperature, humidity, among others)
	3. Source of data (rooms, a cup of coffee, refrigerator, ground, among others)
Decision makers / Action triggers	1. Who takes a decision (Ex. Agent, software system)
	2. Circumstances for triggering action
Action Performers	1. What performs action {Non-IoT actuation}
	2. Type of action (call a system functionality, printing, among others)

5.3.8 Catalog for the - IIA-8: Actuation triggered by an individual, based on non-IoT data.



Catalog – Actuation triggered by an individual, based on non-IoT data	
Entity	**Related Information**
Data producers	1. Who collects data {Non-IoT data collection}
	2. What data is collected
	3. Source of data
Data Exhibitors (display)	1. What exhibits data (Ex. Devices running user applications)
	2. Data format
Data consumer and Action trigger (human)	1. Who accesses data (Ex. Person, Persona, Profile, Role, among others).
	2. Data semantics (the meaning of data according to who visualizes it)
Action Performers	1. What performs action {IoT actuator}
	2. Type of action (Ex. Circular motion, Straight-line motion, On/Off circuit, among others)

5.3.9 Catalog for the - IIA-9: Non-IoT actuation triggered by an individual, based on IoT data



***Catalog* – Non-IoT actuation triggered by an individual, based on IoT data**	
Entity	**Related Information**
Data producers	1. Who collects data {Sensors, Tag readers}
	2. What data is collected (temperature, humidity, among others)
	3. Source of data (rooms, a cup of coffee, refrigerator, ground, among others)
Data Exhibitors (display)	1. What exhibits data (Ex. Devices running user applications)
	2. Data format
Data consumer and Action trigger (human)	1. Who accesses data (Ex. Person, Persona, Profile, Role, among others).
	2. Data semantics (the meaning of data according to who visualizes it)
Action Performers	1. What performs action {Non-IoT actuation}
	2. Type of action (call a system functionality, printing, among others)

5.4 ScenarI$_{oT}$: an IoT scenario description support

We have presented the compilation of IoT-related information previously into catalogs. The information of these catalogs are intended to be captured in scenario description writing, but we conjecture that the path from the selection of the appropriate catalog until the scenario description writing may not be a trivial way ahead, especially for people with little or no knowledge on the IoT background. Based on that conjecture, we propose the **ScenarI$_{oT}$** approach in order to instruct using the Catalogs to write scenarios. The overview of the approach is depicted in Figure 18 and detailed subsequently. A hypothetical scenario will be used as an example in order to assist the explanation. The scenario is “Heart monitoring of patients with heart diseases.”

The path of writing the scenario description can be divided into two main steps: 1) identifying an appropriate IIA (or a combination of them) and 2) employing the correspondent catalog and write the scenario description.

1) ***Identify the appropriate IIA**:* This is the first step in the path before writing the scenario description. This step is concerned with the identification of an appropriate IIA that can abstractly represent the scenario to be described. The IIAs identification is intended to work as a bridge because the IIAs are directly linked with the catalogs, that is, the IIAs are the input to get the proper catalog of IoT-related information.

   This identification can occur in an ad-hoc manner depending on the understanding/experience of the analyst with the IoT background. For example, a

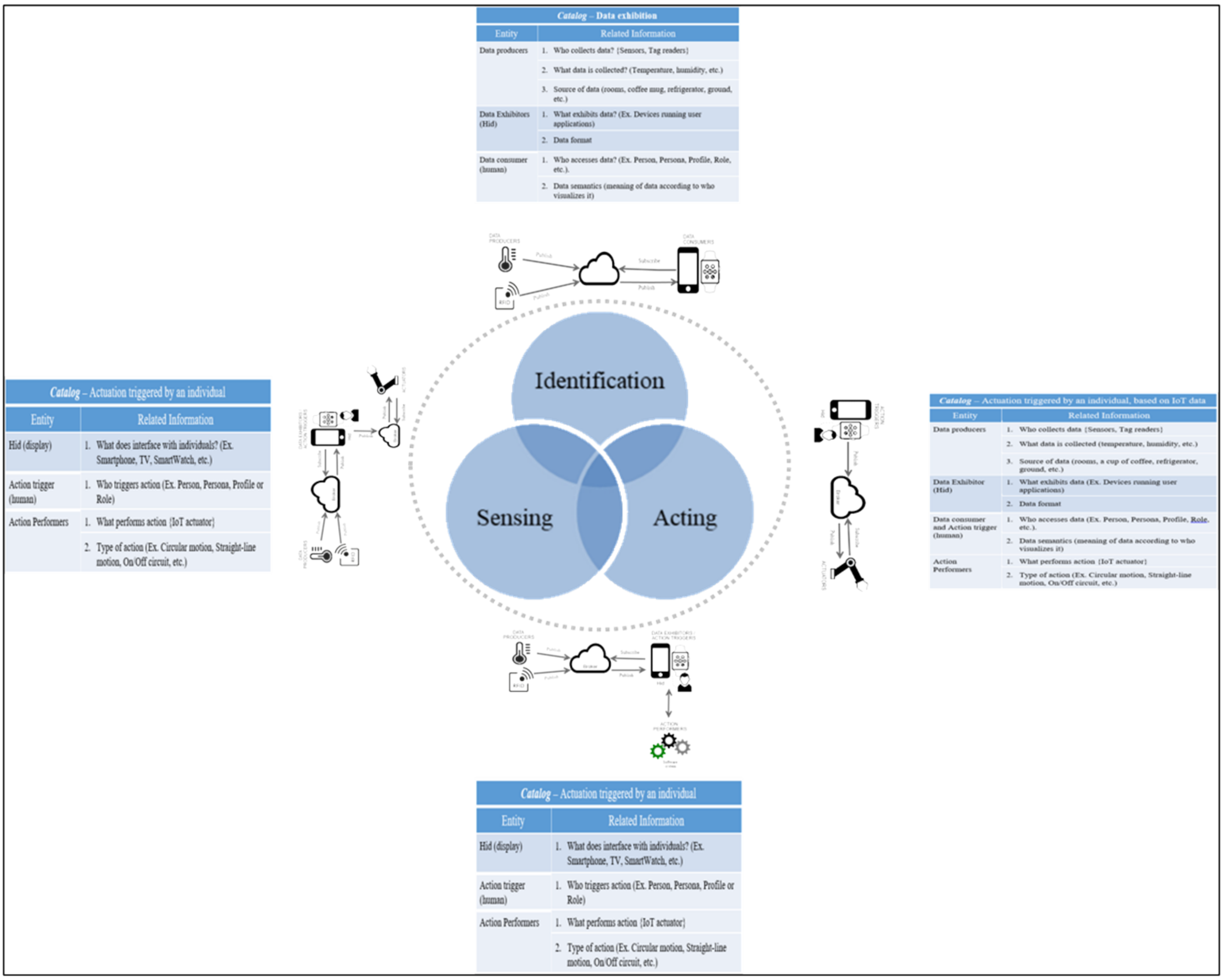


Figure 18 - ScenarI$_{OT}$ approach overview

person who has some knowledge on the Identification, Sensing and Actuation behaviors may not face significant difficulties to decide which IIA is the appropriate one. In this case, the knowledge on the three behaviors might be the input satisfactory to define a relevant IIA (as depicted in Figure 18). For instance, taking our example scenario of heart monitoring, it is tacitly known that the core scenario flow is meant to sense something (produce data) and display the information. It is possible to notice that the sensing behavior is mandatory to realize this scenario; therefore, the IIA-1 - for IoT data exhibition - has been identified as the proper one. On the other hand, some analysts may need step-by-step support for this task.

Considering that one may encounter difficulties in the task of identifying a related IIA, a decision-making flowchart has been designed in order to provide a guide for that task. The flowchart is represented in Figure 19. Numbers have been tagged in some questions for a matter of assisting the explanation. The pre-requisite to start the decision process is to have a scenario at hand to be the input. Once having the scenario, the analyst can follow the instructions provided by the guide to achieve the related IIA (that is if the scenario includes IoT characteristics). If the concerned scenario passes in the question (3) but the data collection and actuation are not IoT-related (questions 4 and 5), then the decision flow ends (6) because the input scenario could not be recognized having the IoT behaviors as a baseline.

Taking the heart-monitoring scenario as an input to exemplify the guide application, the input scenario will pass in the first question (1) “Is there only data collection in the scenario?” but not in the question (2) “Is there only actuation in the scenario?” neither in the subsequent one (3) “Is there both data collection and actuation in the scenario?” It means that the concerned scenario is solely based on IoT data collection, explicitly sensing behavior. Therefore, the decision process ends in the first question and has the IIA-1 as the output.

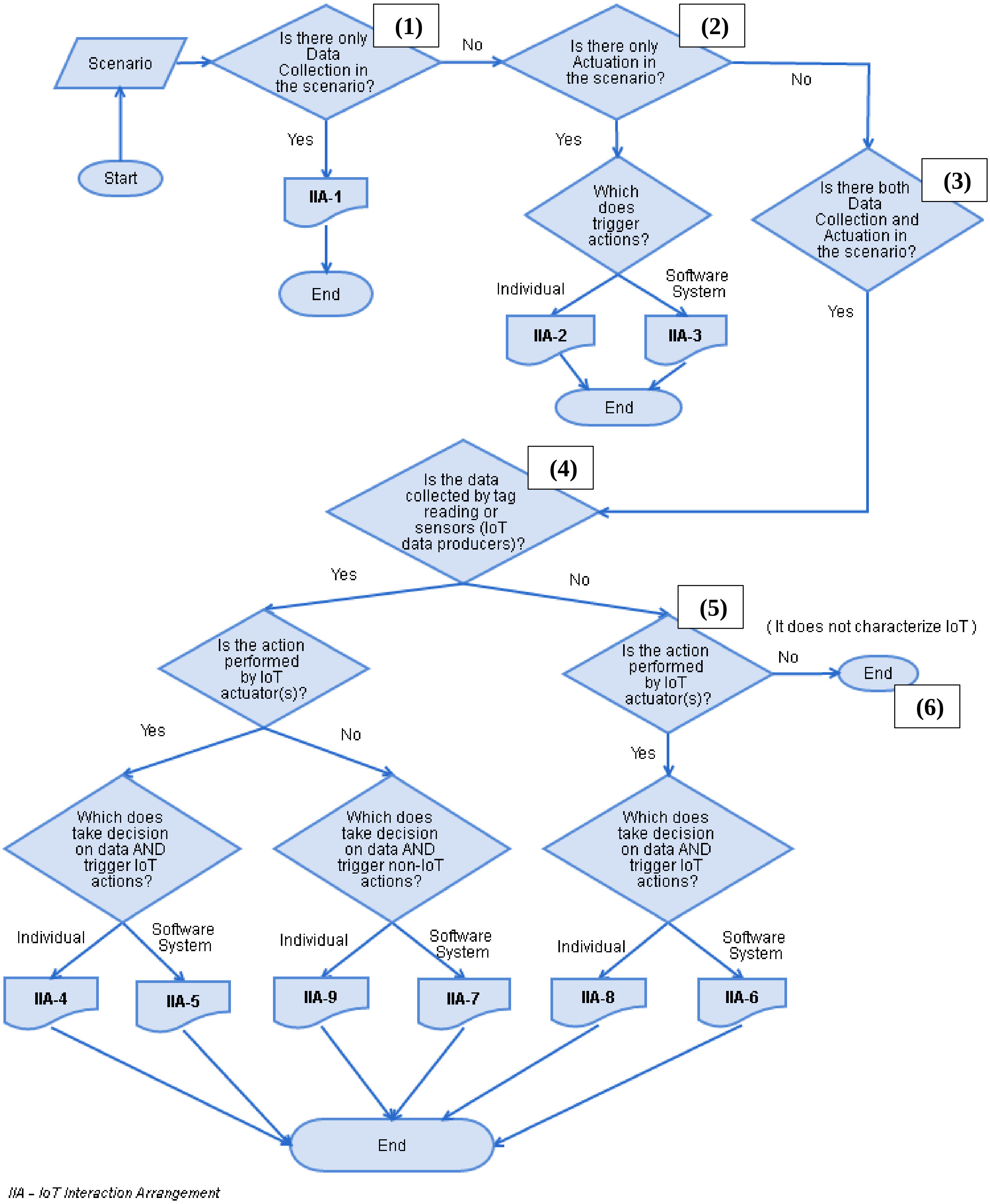


Figure 19 - Guide to support IIA identification

2) ***Employ the correspondent catalog to write scenario description**:* In this step, once the relevant IIA has been chosen, information has to be provided related to the inquiries of the IIA's catalog, and then the scenario description can be written. As the

IIA-1 has been chosen in the first step considering our heart-monitoring scenario, its correspondent catalog has been employed in order to assist us in becoming aware of the IoT-related information that can be captured in the scenario description. As shown below in Table 7, accurate information (colored red) was given to the inquiries of the IIA-1 catalog, and this information was used to fill the scenario description with IoT-related content.

Table 7 - Catalog employment for the heart-monitoring scenario description

Catalog **– Data exhibition**	
Entity	Related Information
Data producers	1. Who collects data? ***Sensor***
	2. What data is collected? ***Heart sound***
	3. Source of data? ***Patient's heart***
Data Exhibitors (Hid)	1. What exhibits data? ***Healthcare professional's mobile devices***
	2. Data format? ***Beats rate per minute***
Data consumer (human)	1. Who accesses data? ***Healthcare professionals***
	2. Data semantic? ***Anomalies or normality in heart rates***

The resulting scenario description (the artifact) was written observing the **ScenarI$_{\mathbf{oT}}$** approach classification set previously. That is, the scenario was written in natural language as shown below. The information raised from the catalog is that underlined in the text. It is also essential to notice that all information raised from the catalog was captured in the artifact following a consistent action flow as expected according to scenario definitions.

The "how-to" scenario is writing itself that is, supporting framing the sequence of actions in the scenario description is not in the scope of this work. It leads us to state a pre-requisite for the **ScenarI$_{\mathbf{oT}}$** approach: the analysts/practitioners in charge of describing the scenarios by employing the **ScenarI$_{\mathbf{oT}}$** need to have previous knowledge on scenario framing.

> ***Heart monitoring scenario description***
>
> *In some hospital, patients with heart diseases are monitored continuously. Sensors capture heart sound from patients' heart and share data to be visualized from mobile devices. Healthcare professionals monitor beats rate per minute in order to observe anomalies or normality in heart rates.*

5.5 Chapter considerations

This chapter presented the procedure of extracting inquiries from IIAs and compiling them into catalogs so that analysts/practitioners can be aware of information relevant to be raised and captured during scenario description activity. That is the brief of the **ScenarI$_{oT}$** approach, which is composed of two significant steps: 1) Identifying the appropriate IIA for a concerned scenario and 2) employing the correspondent catalog to write the scenario description.

We highlight some of the threats to validity related to the **ScenarI$_{oT}$** approach proposal:

- The inquiries of the catalogs were defined in an ad-hoc process. Even though this process was performed observing the objectives stated in the approach classification, some information of the catalogs may not be relevant (considering the purposes), or information may be missing.
- The flowchart guide may have a complex structure, which can bring difficulties to the IIA identification process instead of facilitating it, influencing in the further steps of the whole approach.

The **ScenarI$_{oT}$** approach application was observed through studies carried out in two undergraduate classroom projects. The next chapter presents the protocol of these studies as well as the results.

6 Assessment Studies

This chapter presents the studies carried out in order to assess the ScenarI$_{oT}$ approach version presented in the previous chapter. Two studies were performed in two different classroom projects to observe the employment of the ScenarI$_{oT}$ approach. A more detailed discussion about the assessment results and threats to validity are presented.

6.1 Introduction

The ScenarI$_{oT}$ approach proposed in this dissertation were employed in two projects part of distinct undergrad courses and at different universities, whose students did not have the same degree of experience and knowledge. These two projects have been selected by convenience, mainly because they were part of undergrad courses with a background on IoT. Although they were related to different courses, they shared a similar project theme: shrimp farming management.

For a matter of justification, the theme adoption similarity is explained by the fact that a Brazilian private non-profit company that provides services to support micro and small enterprises had published an article on business opportunities that revolve around shrimp farming. The article reports the need for a management system to monitor a large number of variables that must be continuously monitored in farming. This business gap explains then the adoption of that theme to be addressed in the classes where our studies were performed.

In one class, an IoT-bSS called "Camarão Iotizado" has been delivered. In the other class, students were divided into groups where each one developed its project. The studies are described separately in the next subsections. This chapter finalizes with an overall discussion on the results obtained from the assessment studies as well as the threats to validity observed during the ScenarI$_{oT}$ approach usage and other relevant considerations.

6.2 Assessment study I: *Camarão Iotizado* project

The first study was carried out through the development of a bachelor's undergrad project with a pre-requisite to employ IoT technologies. The IoT-bSS developed in the project was called *Camarão Iotizado* (IoT Shrimp), a system to support shrimp farming. A protocol was defined and followed in the study execution. The detailed protocol is available in **Erro! Fonte de referência não encontrada.**.

6.2.1 Study planning

This study is an initial evaluation of $ScenarI_{oT}$ approach in the perspective of its utility in early-phases of IoT projects. The study's goal is mainly to observe the description of scenarios in IoT-bSS development by using the IIAs and their corresponding Catalogs of information.

6.2.1.1 Research questions and observation points

The research questions are described as follows:

Main question: The description of scenarios based on IIAs and their Catalogs is feasible to support the specification of software requirements in IoT-bSS?

Secondary questions (SQ):

1. [Completeness of arrangements] The set of IIAs is complete from representativeness of interaction possibilities among elements of the IoT context?
2. [Utility of the IIAs] Is the set of interaction arrangements useful for the description of scenarios?
3. [Suitability of catalog information] Is the information about each Catalog relevant and sufficient for the description of the scenarios?
4. [Utility of the Guide for IIA identification] Is the *Guide for IIA identification* useful?

Based on those research questions, we have stated the following observation points:

What to observe?

- ***About SQ1:***
  - Scenarios associated with at least one IIA
  - Scenarios not associated with any IIA
- ***About SQ2:***
  - Perception of developers on the IIA usefulness
- ***About SQ3:***
  - Perception of the developers about the usefulness of the guide to support IIA identification.

    This perception is intended to be measured from a Visual Analog Scale (VAS), a method that has been used in the social and behavioral sciences in order to measure subjective phenomena [87]. The VAS in the questionnaire is a horizontal straight line with 10cm, and the end anchors are labeled with “useless” on one end and “useful” on the other so that participants could inform their perception of utility on the guide.

- ***About SQ4:***
  - [Relevance]
    - Information provided in the catalog considered irrelevant by the developer
  - [Sufficiency]
    - Information provided in the catalog is sufficient
    - Relevant information to the scenario description not captured in the catalogs.

6.2.1.2 Participants

The students enrolled in the undergrad course in which the study was intended to be executed. They are undergraduate students from different fields, Computer Engineering and Electronic Engineering. Students had no knowledge of IoT concepts but a vague reference on the term. Two of them had knowledge of *Arduino* and *Raspberry* development platforms.

6.2.1.3 Tasks

The tasks the participants should execute were defined based on the steps of the ScenarI$_{oT}$ approach for scenario description. The prerequisite for the scenario description activity is to have previously a scenario or a list of system scenarios which are composed of elements from the IoT background.

Tasks:

T1. Find the IIA that represents the scenario to be described (from an *ad-hoc* manner or supported by the *Guide for IIA identification*);

T2. Once the IIA has been found, the corresponding catalogs are filled out.

T3. Finally, the scenarios are described capturing the information from the catalogs.

6.2.1.4 Evaluation instruments

This section defines which resources were used during the study. Three main instruments were designed and can be verified in the study protocol (**Erro! Fonte de referência não encontrada.**):

- A document containing the IIAs and their respective Catalogs;
- A document presenting the *Guide to support IIA identification* (equivalent to Figure 19);
- A *post mortem* questionnaire was designed in order to obtain information from the participant's perception so that it could support answering the research questions.

6.2.1.5 Pilot study

A pilot study was conducted in May 2018, aiming to identify flaws in the planned tasks and at the designed instruments. Three participants were selected by convenience for this trial, where two of them were second-year master students, and the other one was a student finishing a bachelor's undergrad in Computer Engineering.

For this pilot, nine hypothetical scenarios were stipulated to fulfill the pre-requisites. They were designed envisioning possible scenarios for an IoT system to a clinical analysis laboratory. The number of scenarios was equally distributed, and we attempted to assign scenarios with different IoT behaviors for each participant so that they could have contact with maximum behaviors. The nine scenarios of this pilot study are annexed in the protocol.

The pilot session started with a brief explanation on scenario description in requirements engineering followed by a presentation of IoT concepts and the $\text{ScenarI}_{\text{oT}}$ approach. Then the participants performed the T1, T2 and T3 tasks, finishing the session by answering the *post mortem* questionnaire. The session lasted about 45 minutes, including the questionnaire answers.

Regarding the instruments, the participants informed that they were able to perform the three tasks and to answer the questionnaire with no relevant issue. However, regarding the approach itself, one of the participants (Participant A) provided an answer for question 2 of the questionnaire which was an object for analysis. The Participant A suggested an IIA representing a Data Base entity among Data Producers and Data Exhibitors so that "*it could be exhibited not only current/momentary data but also providing data based on the mean or standard deviation calculations*."

We have discussed that suggestion and concluded the following: representing an entity such as a Data Base would increase the IIAs' level of detail, which is not compatible with this work's scope. Further, it has raised another reasoning in which collected data could suffer some processing before being shared for an exhibition, or even some decision could be taken before data sharing. These could be responsibilities of a Software System abstract entity just as in the IIA-5 and IIA-7. However, at that moment we understood that it was similar to the Data Base concern. That is, it is a finegrained matter that is not compatible with this work's goals, but it can be an issue to be addressed in future works.

6.2.2 Study execution

This observational study was executed during the course, but the tasks related to this research was performed among May and June 2018. The project did not count with a specific client or stakeholders in which functional requirements could be elicited. Instead, in the early phases of the project, the students performed brainstorming activities in order to idealize and elect the systems' features based on available materials such as videos, magazine articles and reports from Brazilian government institutions. In summary, the primary system goal defined was *to monitor tanks for shrimp farming keeping interested people aware of indicators concerning the water level, temperature, turbidity, and salinity*.

From the primary goal, the students defined the system's features and the actors involved. From these features, the students identified the scenarios to be described, which were intended to complement the system requirements specification. Four scenarios with an IoT background were identified, as follows:

1) Tank keeper receives notification of adverse situation in the shrimp tank
2) The manager receives notification of adverse situation in the tank
3) Tank keeper visualizes tank Information
4) Manager visualizes the tank information panel

As the pre-requisite for the project development was to employ IoT technologies, then an IoT lesson was given so that they could understand the IoT paradigm, analyze and design the solution. Further, a brief explanation of the scenario description in requirements engineering was given as well as a presentation of the ScenarI$_{oT}$ approach.

There were six students enrolled in the course, and three groups were divided for the high-level activities of (1) specifying systems requirements, (2) developing the system *front-end* and (3) developing the *back-end* (including setting up the microcontroller kits, sensors, etc.). One group of two students were responsible for the activity (1), and, therefore, they performed the scenario specification activity.

The group was asked to perform the T1, T2, and T3 steps to describe the scenarios. Then, the students answered the *post mortem* questionnaire in conjunction. The resulting scenarios, as well as the results of the study, are presented in subsequent.

6.2.3 Results

The participants described four scenarios (Tables 8, 9, 10, and 11). In the T1 task, the Guide was used in order to support them in identifying the pertinent IIA. Two scenarios were related to the IIA-7, whereas the remaining were considered instances of the IIA-1.

In the scenarios related to the IIA-7 (Tables 8 and 9), the participants captured all the Catalog information. Just for a matter of facilitating the reading and analysis of information on the artifacts, we have highlighted with distinct colors portions of text related to each type of information, that is, information of Data Producers (highlighted in green), Decision makers/Action triggers (highlighted in blue), and Action performers (highlighted in yellow).

Table 8 - Scenario description 1

Scenario 1: Tank keeper receives notification of adverse situation in the shrimp tank
The sensors in the tank collect data regarding water level, temperature, turbidity, and salinity. The data collected by the sensors in the tank are sent to the system and processed. The system checks if any of the data is in non-compliance with the previously set reference value. If any of the data is in non-compliance, the system sends to the tank keeper a notification by e-mail or internal signaling specifying which of the data is in non-compliance and what type of anomaly it is, in addition to the current value recorded by the sensor.
IIA-7

Table 9 - Scenario description 2

Scenario 2: Manager receives notification of adverse situation in the tank
The sensors in the tank collect data regarding water level, temperature, turbidity, and salinity. The data collected by the sensors in the tank are sent to the system and processed. The system checks if the values recorded by the tank sensors have a relative discrepancy of more than 10% against the previously set reference value. If any of the data has a relative discrepancy more significant than 10% against the reference value, the system sends the manager an e-mail notification or internal signaling indicating which of the data is in non-compliance and what type of anomaly is recorded.
IIA-7

Unlike previous descriptions, the participants did not capture all the information from the IIA-1 correspondent catalog. They opted not to capture any information of Dada Exhibitor entity, that is, to explicit the “data format” and “what exhibits data.” As can be verified in Tables 10 and 11, the participants captured information of Data Producers (highlighted in green), as well as information of Data Consumers (highlighted in gray).

Table 10 - Scenario description 3

Scenario 3: Tank keeper visualizes tank Information
The sensors in the tank collect data regarding water level, temperature, turbidity, and salinity. The data collected by the sensors in the tank are sent to the system and processed, enabling them to be visualized. The tank keeper enters the information display panel. The system provides data visualization, which allows the tank keeper to check if there is a need to make provisions in order to change the situation of the tank.
IIA-1

Table 11 - Scenario description 4

Scenario 4: Manager visualizes the tank information panel
The sensors in the tank collect data regarding water level, temperature, turbidity, and salinity. The data collected by the sensors in the tank are sent to the system and processed to enable them to be visualized. The manager enters the information display panel. The system provides data visualization, which allows the manager to assess the overall situation of the shrimp farm.
IIA-1

The two students answered the *post mortem* questionnaire in conjunction (the questionnaire is annexed in the study’s protocol). In the answer for Q1, they confirmed they were able to associate each of the four scenarios with at least one IIA. They employed the Guide in other to assist the identification.

For Q2 they informed it was not possible to identify the existence of another arrangement. In the Q3 they reported two redundancies in the set of nine IIAs. The first one is related to IIAs 2, 4 and 8. The participants reported that “*differentiation between why the individual (human) triggers a particular action is not relevant enough at a system level for different arrangements to be needed*.” They also found a redundancy among

IIAs 3, 5 and 6, arguing that "*differentiation between why another software system triggers a particular action was perceived to be outside the scope of the arrangements definition, and therefore the existence of different arrangements for these actions was considered unnecessary*."

Related to the Guide utility, considering the questions 4 and 5, they informed that the Guide was useful to assist identifying IIAs, and their utility perception measured score pointed out in the VAS was 8.2.

In the Q6 participants confirmed they employed the Catalogs as a basis to describe the four scenarios. Also, on the relevance of Catalogs' information, they reported that "*information questioned by the Catalog served as support*" (Q6.a) and they did not need to capture information that was not provided from the Catalogs (Q6.b).

Moreover, the participants left other explicit comments and suggestions. They commented on the relevance of $ScenarI_{oT}$ approach when said: "*the use of the IIAs, Catalogs and the support Guide facilitated the construction of the scenarios, mainly due to the lack of contact with IoT background before the course*." However, they left a suggestion on the scenario description artifact itself. It was considered that "*the resulting scenario descriptions could have been clustered within use cases artifact, creating special use case formats for IoT, which was how we used the artifacts*." Lastly, one participant that integrated the group responsible for back-end and microcontroller kits responsibilities has commented that the IIAs "*could be used to support system architecture design in future projects*."

6.2.4 Research questions answering

Considering the results obtained from this study, we can answer the research question, starting from the secondary ones.

- **SQ1.** On the completeness of arrangements, we could observe that scenarios were associated with at least one IIA. However, we understand the fact that these four scenarios identified in the project were not diverse; it made challenging observing scenarios that could not be related to any other IIA. Although the participants informed they could not identify the existence of another IIA (answer to the Q2 of the questionnaire), they could be biased by the limited scope of the project and also due to the lack of in-depth knowledge on the IoT background. Therefore, by considering the context of the study, that is, the small set of scenarios that could be associated to IIAs, we can answer that the set of IIAs is complete in that context.
- **SQ2.** Participants have reported redundancies in the IIAs. On these redundancies reported, we understand the lessons given on the Scenario-IoT

approach before executing the study could not have adequately reflected the reasons why we had designed IIAs in which individuals (humans) trigger actions and why there are IIAs where Software Systems entities trigger different actions (IoT and non-IoT actions). Although these redundancies were reported, they have expressed that IIAs were useful together with the Catalogs and the Guide to support IIA identification. For this reason, we understand that that IIAs were useful for the scenario description task.

- **SQ3.** The answers to the questionnaire of Q4 and the VAS score of Q5 confirm that the Guide (the flowchart) to support IIA identification is useful.
- **SQ4.** On the relevance of Catalogs information, although participants did not make explicit “data format” and “what exhibits data” in the description of scenarios related to IIA-1, we observed they meant to capture that in later phases. Considering the answers Q6, Q6.a and Q6.b, we understand the information about each Catalog was sufficient for the description of the scenarios but not relevant as information was omitted in that phase of development.

Considering the points that have been observed and the answers to the secondary questions, we can answer the main research question of this study arguing that the description of scenarios based on the $ScenarI_{oT}$ approach was feasible to support the specification of software requirements in the *Camarão Iotizado* software system. For clarification purposes, the scenario artifacts complemented the software requirements specification in this project; that is, the developers did not base only on the scenarios to analyze and specify the system. As participants reported, the scenarios artifacts were blended with use cases description artifact and therefore used in conjunction.

6.2.5 Threats to validity

At the limited scope and population: The study was carried out with undergraduate students attending an IoT-related course. The study counted with only six participants. Besides, the scope of the project was minimal, and for this reason, some points could not be observed such as that related to the existence of another IIAs that could not have been captured in this work development.

At participants imprecision on reporting: As students composed the population of the studies, they could have been prevented from issues (such as deadlines, lack of motivation and also the need to obtain good degrees) to provide more precise or reasonable answers when reporting perceptions and answering the questions of the questionnaire.

At the absence of IoT specialists: The assessment study was carried out in undergrad project. It is important to emphasize that the lack of IoT specialists as participants of the study is a threat to validity for this research.

6.3 Assessment study II

This study was carried out in a *Programming Topics* undergrad course in a different university of the previous study. This course counted with a higher number of enrolled students compared to the course of the first. The students were divided into seven groups so that each one could develop its solution for the *Shrimp Farming* problem (the reason why the themes are similar was explained in this chapter's introduction). One important note is that the project's development had already begun when we obtained the appropriate permission by the responsible professor to execute the study.

Considering the higher number of students and that more than one project was expected to be developed, we planned this study, attempting to keep some observation points of the previous study but also to append other ones. We conjectured that the IIAs and the Catalogs might have the potential to support other activities of the development life cycle, not only the scenario description itself. This conjecture was reinforced by the comments obtained in the study I. For this reason; we have planned the present study as described in the next subsection (a protocol was defined and followed in the study execution, which is available in **Erro! Fonte de referência não encontrada.**.

6.3.1 Study planning

This study aims at evaluating the $\text{ScenarI}_{\text{oT}}$ approach in the perspective of its utility but also exploring other possible utilities of the IIAs and the Catalogs in development activities.

6.3.1.1 Research questions and observation points

The research questions were stated as follows:

Main question: Are IIAs and Catalogs useful for IoT-bSS development activities?

Secondary questions (SQ):

[Interaction arrangements]

SQ1. What are the developers' perceptions about the usefulness of the IIAs for each activity in which they were applied?

SQ2. What is the contribution of using the IIAs for each activity?

SQ3. Is the set of IIAs enough to represent the different possibilities of interaction ("things," systems and actors) of the projects developed?

[Guide for IIA Identification]

SQ4. What is the perception of the developers about the usefulness of the Guide for the identification of IIAs?

[Catalogs]

SQ5**.** What are the developers' perceptions about the usefulness of the Catalogs for each activity in which they were applied?

SQ6**.** What is the contribution of using the Catalogs for each activity?

SQ7**.** Is the information in each Catalog relevant and sufficient?

What to observe?

1) ***About SQ1:***
    - Perception of developers regarding the use of IIAs for activities where they were applied, such as:
        - Discovery of functional and non-functional requirements
        - Definition of the interaction architecture of system components
        - Creation of behavioral and structural models
        - Other activities.
2) ***About SQ2:***
    - Contributions of IIAs for the activities performed.
3) ***About SQ3:***
    - Scenarios associated with at least one IIA.
    - Scenarios not associated with any IIA.
4) ***About SQ4:***
    - Perception of developers regarding the use of the Guide.
5) ***About SQ5:***
    - Perception of developers regarding the use of the Catalogs for the activities where they were applied, such as:
        - Description of IoT scenarios
        - Creation of behavioral and structural models
        - Other activities.
6) ***About SQ6:***
    - Contributions of the Catalogs to the activities carried out

7) About SQ7:

[Relevance]

- Information captured in the Catalogs considered irrelevant by the developer

[Sufficiency]

- Information provided in the catalog is sufficient
- Information relevant to a particular scenario description not provided for the Catalogs.

6.3.1.2 Participants

Not all students enrolled in the course have participated in the study. Also, the study was divided by observing the group's activities and individual activities. From the seven formed groups, three contributed entirely to the study - named Alfa, Beta, and Gama. The Alfa and Gama groups were formed by four members, whereas the Beta group was integrated by five. Almost all the participants were advanced students. As shown in the graph of Figure 20, the participants of group tasks were six-year and seven-year students. On the other hand, eighteen participants contributed entirely to the

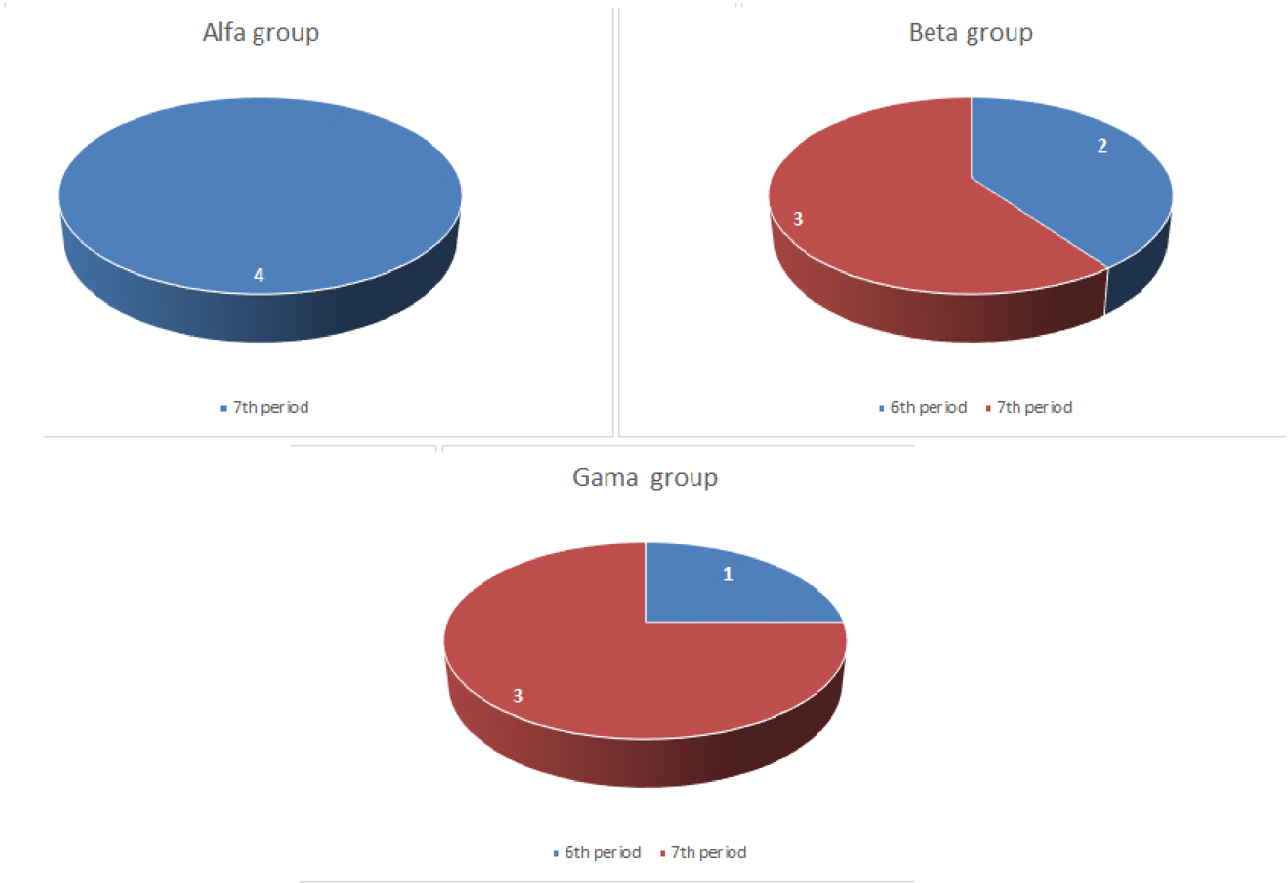


Figure 20 - Groups' distribution of students by period

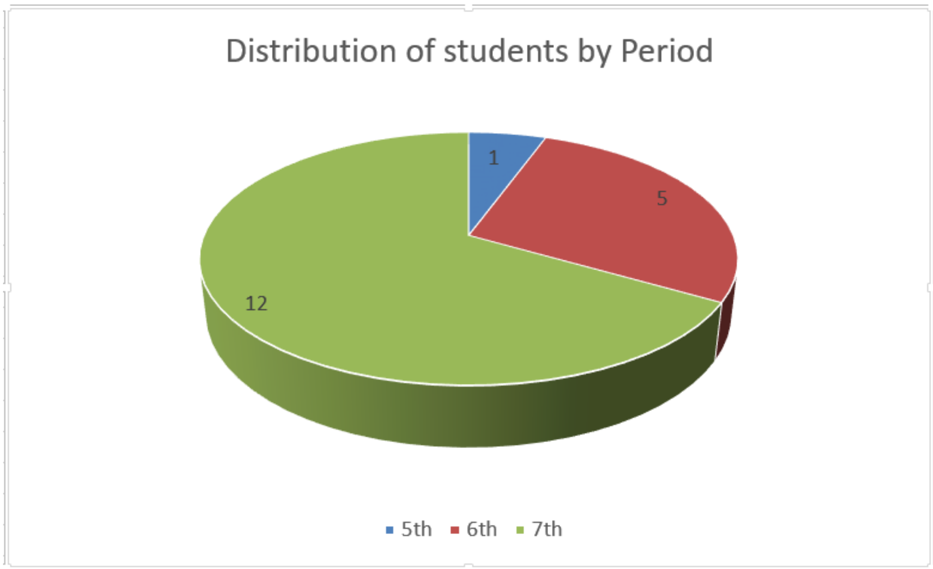


Figure 21 - Distribution of students by period (individual task)

individual task. They were five-year, six-year and seven-year undergrad students (Figure 21), with no prior knowledge on IoT concepts, but some of them had been already developed small projects employing the platforms *Arduino* and *Raspberry Pi*.

6.3.1.3 Tasks

As explained before, the tasks the participants should execute were divided into the ones performed in groups and the other performed individually.

6.3.1.3.1 Group task

The groups were asked to develop their projects to support shrimp farming. As noticed, the projects were already in advanced phases when we started the study execution. For this reason, the $ScenarI_{oT}$ approach was only suggested to be applied as a complement to the systems requirements specification. That is, it was not mandatory to perform scenario description as there were projects' deadlines and it could then interfere with the projects' progress. Also, participants were suggested to attempt employing the IIAs and the correspondent Catalogs in whatever project activity they eventually considered pertinent (not being mandatory as well). From these two activities, we could observe the seven points stated.

The only necessary task was that in which the groups should associate the scenario(s) of their systems to the IIAs they considered being relevant. From this task, we could observe point 3. In summary, group tasks are:

Tasks:

T1. Associate the projects' scenario(s) to related IIAs (mandatory)

T2. Describe scenarios by employing the $ScenarI_{oT}$ approach (not mandatory)

T3. Employ IIAs and Catalogs in further development activities (not mandatory).

6.3.1.3.2 Individual tasks

In other to assist observing the points 3 and 4 (regarding IIAs association to scenarios and the utility of the Guide), we planned an activity in which participants should associate IIAs to scenarios from hypothetical features with an IoT background. The task was asked to be performed by two manners:

Task:

T4. Interpret hypothetical features of a "Smart City" and identify scenarios and the IIA(s) that could be related. The IIA identification should be performed by an *ad-hoc* manner, followed by the identification employing the Guide (in this sequence). This activity aims at assisting us in observing the Guide utility. Further, in other to assist in answering the main research question, we asked participants to express their perceptions on development activities in which the IIAs or Catalogs could have the potential to be employed in future projects.

It is essential to highlight that the hypothetical features were written in a manner that it could be divided into sub-scenarios depending on each interpretation. Besides, the features description did not present concrete information such as types of sensors or actuators. The reason is also to take advantage of participants' interpretations that could result in different IIAs associations. The features description of the Smart City is this below:

> "Keep the Wi-Fi signal and the reservoir (water fountain) of the main square connected after 2:00 am only when citizens are passing through the area at this time. In the case of the reservoir, it must be switched on regardless of the time only if the level of the dam that supplies the city is above average."

6.3.1.4 Evaluation instruments

This section defines which resources were used during the study. Three main instruments were designed and can be verified in the study protocol (**Erro! Fonte de referência não encontrada.**):

- A document containing the IIAs and their respective Catalogs;

- A document presenting the *Guide to support IIA identification* (written in Portuguese);
- A form for a semi-structured interview was designed in order to obtain information from the groups' members so that it could support answering the research questions.
- A document containing possible features of a Smart City and the particular questions to accomplish the individual task.

6.3.2 Study execution

This observational study was executed in June 2018. Before the study application, the students had performed brainstorming activities in order to idealize and elect the systems' features. In summary, all groups dedicated to the sensing behavior, and the projects followed a similar goal: *to monitor tanks for shrimp farming, keeping interested people aware of indicators concerning related variables*.

Similar to the first study, lessons were given to students to explain what scenarios are, and also to present IoT concepts and the elements that compose the $\text{ScenarI}_{\text{oT}}$ approach, that is, the IIAs, the Guide to support IIAs identification and the Catalogs of information. In this lesson, the students were asked to raise the systems' scenario(s) and associate those to the IIAs previously explained. They were suggested to attempt describing the scenarios by using the $\text{ScenarI}_{\text{oT}}$ approach and also to attempt employing the IIAs or Catalogs in the development activities they would perform.

From the systems' features, the students identified a scenario or a set of them and associated the IIAs (***task T3***). The students presented the results in a classroom session. Besides associating IIAs, some groups also incremented/tailored the original representations with more accurate information. An example is the one depicted in Figure 22 and Figure 23. All seven groups concluded this task.

After delivering the projects, we applied the semi-structured interview with the groups' members in conjunction. Then, the present students were asked to perform the

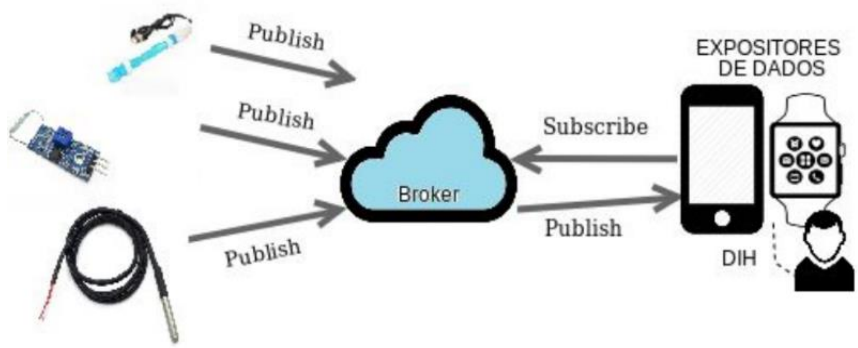


Figure 22 - Tailored IIA b)

task T4. That is, not every student enrolled in the course has contributed to the individual task. In summary, as depicted in Figure 21, eighteen students were present in the day where the individual task was intended to be applied. The participants were asked to interpret the scenario of a smart city and then answer the questions in the right sequence so that they could perform the *ad-hoc* IIA association before using the Guide to the same matter. The objective was avoiding biases from the Guide usage.

6.3.3 Results

The presentation of the results will be organized according to the tasks performed.

6.3.3.1 Task T1 results

All seven groups have performed this task. The IIA-1 (Data exhibition) was chosen by all groups to represent their scenarios because they have stated the same scenario: *Tank monitoring by the shrimp farming manager*. The main difference among the projects was the sensors used to collect data, implying in different types of data collected from the tanks (e.g., water temperature, dissolved oxygen, salinity, transparency, alkalinity, ammonia, nitrite, etc.), but this issue does not influence the IIA association.

It was not possible to collect the Guide perception utility from all groups. As said before, only three groups have contributed entirely to the study (the Alfa, Beta, and Gama), and these have informed their perception of the Guide utility. Three members of the Alfa groups expressed their perception, resulting in a VAS average score of 9.4. From the Beta group, the average score was 8.5 (one member did not answer). Only two members of the Gama group informed their perception, resulting in an average score of 10.

Figure 23 – Tailored IIA a)

6.3.3.2 Task T2 results

No groups have performed this task.

6.3.3.3 Task T3 results

The groups Alfa, Beta, and Gama, have employed IIAs (not the Catalogs) in further development activities. These three groups passed by the semi-structured interview and the results are listed below:

- Question 1: The three groups declared they employed IIAs in the activity: "Definition of the interaction architecture of system components."

- Question 2: The reported utilities of IIAs for the activity reported in question 1 were:
  - Group Alfa: *"IIA facilitated understanding for beginners on IoT and analogy; useful since it became easier to understand the system, what are the actuators and data collectors; we had a vision of how the system could work and its goals; from IIAs, even an outsider will easily identify what the system is intended for".*

  - Group Beta: *"Arrangements helped identifying project structure; view requirements; delimit requirements; idealize the physical part together with the logical part; design of a project with no specification; visualize the communication between components."*

  - Group Gama: "*A priori did not know how to represent the system; The arrangements served to represent to other developers or stakeholders how we thought about the architecture; representation for more lay people*."

- Question 3: As described before, the three groups employed the IIA-1 (Data exhibition). No group has reported the existence of any other IIA or even possible redundancies and combinations among them.

- Questions 4, 5 and 6: No groups have employed the Catalogs in their projects. For this reason, there were no answers to these questions.

6.3.3.4 Task T4 results

It is the individual task. Twenty-seven students performed this task, but only eighteen completed or gave answers with any consistency. For instance, even after analyzing the participants' interpretations, some IIAs association to the smart city scenario did not make any sense. We emailed some participants in an attempt to obtain further explanations, but we received no answer. The fact that the scenario was opened to interpretation may be a threat to validity.

6.3.3.4.1 Association of IIAs to the smart city scenario

Considering the objective of this task (assisting us observing the Guide utility), we have organized the results by (1) participants that identify the same IIA in both *ad-hoc* and supported methods; (2) participants that obtained incompatible identifications in both methods; (2.1) participants that added IIAs by using the guide; and the opposite, (2.2) participants that removed IIAs by using the guide; finally, (2.3) participants that provided totally different answers considering the two methods. The results considering the answers of the eighteen participants are the following:

1. Eleven participants (61.1%) have chosen the same IIA in both methods (*ad-hoc* and supported by the Guide);
    - The most IIA chosen was the IIA-5 with eight occurrences. This choice can be explained by the fact participants interpreted the smart city features in a manner that "*the software system would receive the sensed data of individuals presence and water level, and will act through actuators to turn on and off the reservoir and the Wi-Fi*."
    - Another interpretation led participants to break the features into two scenarios and to choose IIAs 5 and 9 in conjunction. It achieved three choices. It can be resumed and explained by one participant's comment: "*IIA-5 because the use of a sensor to sense the reservoir, and actuators controlled by the system to release the water; Besides IIA-9 because sensors can be used to sense individuals presence, where data is sent to the system that then checks whether the time corresponds with the requirement and the system releases the Wi-Fi signal to people*".

2. Seven participants (38.9%) have chosen different IIAs.
    2.1. From the previous results of b), only one participant has added an IIA in the supported method. This participant had chosen the IIA-9 in the *ad-hoc* method, and then add the IIA-5 when using the guide. This participant has led the

following explanation: "*After using the guide, it became clear that the arrangement 5 is ideal for the functionality of the reservoir because it presents the use of actuators. For the Wi-Fi, the ideal would be the use of arrangement 9 because it presents a system making decisions and acting according to decisions*".

2.2. Three participants have removed IIAs after using the Guide.

- First case: In the *ad-hoc* method the participant has chosen the IIA-5 and IIA-9 with the following comment: "*Arrangements 5 and 9 have been chosen because it will require presence sensors in the square and sensors measuring the water level of the dam, actuators in the reservoir and to turn Wi-Fi on / off if it is in accordance with the sensor data and the functional requirements of the project*". However, when the Guide was used the participant chose only the IIA-5 and left this comment: "*The guide helps to determine which arrangement fits best by inputting inquiries about how the project works (behaviors) to solve the problem*";
- Second and third cases: Similar to the first case, the IIA-5 and IIA-9 were used in the *ad-hoc* method followed by the IIA-5 choice by using the guide. However, these participants have not made explicit the reasons why they modified their choices.

2.3. Finally, three participants have provided different answers, considering the two methods.

- First case: The participant has chosen IIA-5 then IIA-3. However, no detailed explanation was given related to the changing.
- Second case: In this case, the participant has reported many interpretation options. In the *ad-hoc* method, he/she chose the IIA-5 AND IIA-3 OR IIA-5. In the supported method, the IIA-5 was kept, but IIA-3 was replaced by the IIA-7. The explanations for the *ad-hoc* result was: "*IIA-5 to turn on/off the Wi-Fi because the sensors would provide the information to a system that would verify the conditions and then trigger the actuator to turn on the Wi-Fi. In the first (ad-hoc) question, this arrangement seemed to be the closest to what I was looking for, and it was confirmed in the following question when applying the Guide. Further, IIA-3 or IIA-5 to turn the fountain on / off. I chose 3 because I did not bother to collect data. Alternatively, 5 in case the state of the dam was verified by IoT sensors. Therefore there is actuation and data collection by actuators and IoT sensors*".

On the other hand, the explanation given for the supported method results was: "*Changed arrangement 3 to 7 based on the guide. Arrangement 7 was complete and specific for the case*".

- Third case: The participant has chosen IIA-5 then IIA-7. However, no detailed explanation was given related to the changing.

In this task, all participants have reported their utility perception on the Guide. **The average value from the VAS scores was 8.83.**

6.3.3.4.2 Reports on IIAs or Catalogs utility perception for other activities

A procedure was performed based on the textual analysis in which codes were assigned to activities considering portions of data in order to assist us in identifying patterns from similarities and differences emergent from the data obtained. Eight codes have been assigned as depicted in Figure 24. These codes represent the possible IIAs utilities reported by the 18 participants. The utilities were organized into (a) utility for professionals (developers, analysts, etc.); (b) utility for customers/users; and (c) the possible utility for both ones. The results were checked by another researcher specialist in IoT.

6.3.3.4.3 Suggestions and comments

Although no groups have employed the Catalogs in their projects, one participant commented that they had only filled the Catalog but had no employed it. This participant said that felt "*the field data source confusing*." Another participant left a comment on Guide usage. He/she said that "would be better whether there was in the Guide an explanation on the output IIAs."

6.3.4 Analysis

In an attempt to validate analyze the answers provided by the participants on the activities they had employed the IIAs (group tasks), we presented the results to the professor responsible for the course, and he gave his perception on the results.

6.3.4.1 Group tasks results

According to the results, all groups reported they used IIAs in the activity of "Definition of the interaction architecture of system components." The professor raised the possibility that the students could not necessarily have used the IIAs for the architecture definition in a whole. The students had already the first solution thought for the systems before the study execution. They could have attempted to fit their tacit solutions into some category (IIA) that was considered compatible. The professor agreed with the fact the IIAs could have formally supported the architecture definition because there was no architecture documentation before the lesson presenting the IIAs.

On the answer of some member of group Beta related to IIAs helping viewing and delimiting requirements, the professor explained that there were already requirements defined that revolved around the type of data that should be sensed from the shrimp farming tanks. However, after being presented to the IIAs, the professor observed that students could idealize new features that were not thought before. As it was not a current project – where requirements are defined by a specific stakeholder – then there was the possibility to idealize features for the systems. In summary, "*students could actually have been inspired by the arrangements to have insights on features; they could have expanded the system scope after the IIAs presentation*."

6.3.5 Research questions answering

Based on the results and analysis we could answer the research questions as follows:

SQ1. What are the developers' perceptions about the usefulness of the IIAs for each activity in which they were applied?

Answer: Developers felt the IIAs useful for the activity of "Definition of the interaction architecture of system components."

SQ2. What is the contribution of using the IIAs for each activity?

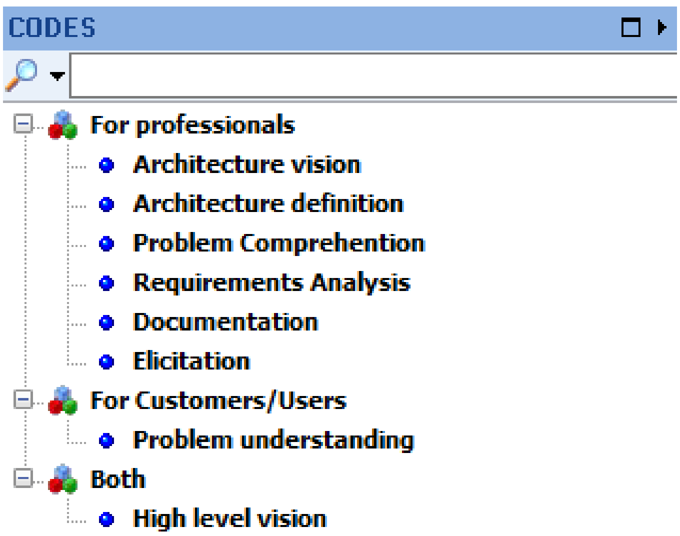


Figure 24 - Assigned codes to the reported IIAs' utilities

Answer: Considering the results and the validation performed with the professor of the course, for the activity mentioned above, the IIAs contributed to the following:

- *Architecture understanding and documentation for beginners on IoT;*

Also, participants expressed other utility that was not directly related to architecture, which was:

- *Features idealization;*

SQ3. Is the set of interaction arrangements enough to represent the different possibilities of interaction (things, systems and actors) of the projects developed?

Answer: Considering the limited scenarios raised in the projects, we could not answer this question. The projects were limited to the monitoring of tanks and, consequently, the participants worked only with the IIA-1.

On the perception of the developers about the usefulness of the Guide for the identification of IIAs (**SQ4**), we obtained positive VAS scores. 9.4, 8.5 and 10 were the utility perception average from the groups when they had to identify the related IIA to the particular scenario of their project. In the individual tasks, we obtained an average score of 8.83. An issue that was observed is that the average score obtained from participants that achieved the same results in the *ad-hoc* and supported methods was higher than the average of that one that obtained different results (8.95 vs. 8.65). In theory, it should be the opposite as people that demonstrated some difficult to identify IIAs by *ad-hoc* would provide higher values for the utility perception on the Guide. However, considering the overall scores, the Guide has indicated to be useful to support identifying IIAs.

The results obtained in this study could not help answering the questions **SQ5**, **SQ6,** and **SQ7** because students had not employed the Catalogs in their projects.

After answering these secondary questions, we could answer the main question.

Main question: Are IIAs and Catalogs useful for IoT-bSS development activities?

Answer: IIAs indicated to be useful for architecture definition and documentation activities, specifically for beginner developers. IIAs could also assist in idealizing system features

6.3.6 Threats to validity

At the limited scope and population: the study was carried out with undergraduate students attending an IoT-related course. It counted with eighteen participants. Besides, the range of projects was minimal, and for this reason, some points could not be observed such as that related to the existence of another IIAs that could not have been captured in this work development.

At participants imprecision on reporting: students composed the population of the study, they could have been prevented from issues (such as deadlines, lack of motivation and also the need to obtain good degrees) to provide more precise or reasonable answers when reporting perceptions and answering the questions of the interview. Also, some participants could not have followed the requirements of performing the task T4 in the correct sequence. It implies them being biased by the Guide. For these reasons, and considering the low number of participants, we could not perform further analysis of the results.

At bias on the semi-structured interview: In other to help the groups' members reporting the activities they employed the IIAs or Catalogs, we have added options of development activities in the interview so that participants could choose one of them or reporting another one. However, we understand those options are threats to validity as they could have biased the answers from the groups.

At the imprecision of the text describing the smart cities' scenario: The scenario description presented for the participants did not carry details about sensing, how data is collected and from which source or how actuation triggering is performed. It is conjectured that this imprecision could allow participants to think of different ways to realize that business problem, being able to slice the scenario as they consider relevant and enabling the use of different IIAs according to various problem realizations. From the purposeful imprecision or lack of details of the scenario description, some participants could break up the scenario in sub-scenarios (pieces) as they considered relevant. Also, they could ponder about different developments/applications for the smart cities' scenario and its parts (when pertinent). Despite exciting developments that raised, the imprecision of the text represents a threat to validity as participants could have felt confused and found difficulties when interpreting the scenario, interfering on the performing of the activity.

At the absence of IoT specialists: The assessment study was carried out in undergrad projects (not real ones). It is important to emphasize that the lack of IoT specialists as participants of the study is a threat to validity for this research.

6.4 Discussion

From the results obtained we could conclude the ScenarI$_{oT}$ approach was useful when applying in the Camarão Iotizado project. Unfortunately, the approach could not be employed in study II. However, we could observe the employment of the IIAs in other development activities, and the results were positive and have raised insights on future works.

One crucial point to be discussed is that, although all studies' threats to validity, it was relevant executing the initial studies during undergrad projects, because it made possible observing that the approach and assets developed in this work can be useful resources for beginner practitioners on the IoT realm taking first steps in the IoT-bSS development. The results have shown from the reports of students that they acquired a more in-depth understanding of the IoT paradigm and also on the possibilities of interactions among the elements about this realm.

Along with the studies, we obtained suggestions regarding the approach. In the first study, the participants opted not to capture information on “data format” and “what exhibits data” in their scenario artifacts. Subsequently, in the second study, a participant reported that he/she got confused with the “data format.” These are valuable suggestions to be deeper analyzed and might be taken into consideration in the evolution of the approach. This is an insight for future work as it was not possible to include the approach evolution in the scope of the present research.

6.5 Chapter considerations

In this chapter, we presented two studies that were carried out in order to observe and assess the employment of the $\text{ScenarI}_{\text{oT}}$ and the assets developed in this work. Although the scope and population limitations and other threats to validity, we could observe a positive use of the approach and the resources that ground it. In the first study, we could observe mainly participants describing scenarios using the $\text{ScenarI}_{\text{oT}}$. In the second study, we focused on the conjecture that the IIAs and Catalogs could be useful for other IoT-bSS development activities.

Although the approach was not empirically experimented, the observation studies provided information that increases the belief in its applicability. In summary, we have observed positive results on the usefulness of the $\text{ScenarI}_{\text{oT}}$ approach to support scenario descriptions and on the usefulness of IIAs to support architecture definition and documentation activities (Catalogs were not employed in other activities). However, limitations are known as the threats to validity presented for the studies performed.

7 Conclusion and Future Works

This chapter presents the conclusions concerning the research questions and contributions generated by this research. Moreover, the limitations of the results achieved are provided, and a few suggestions of future work are proposed.

7.1 Introduction

This work presented the conducted research that aimed at investigating scenario description as a complement for IoT-bSS requirements specification. The main research question of the work is:

How to perform scenario specification of IoT-based software systems so that the IoT domain components and properties are captured?

In order to assist answering this question, we have performed a structured literature review to gather IoT concepts and properties so that we could obtain an in-depth understanding of IoT and realize what information from this realm can be captured in scenario descriptions. From the results, we interpreted IoT as “a paradigm that allows composing systems from uniquely addressable objects (things) equipped with identifying, sensing or actuation behaviors and processing capabilities that can communicate and cooperate to reach a goal.” Also, in our interpretation, “things exist in the physical realm, such as sensors, actuators and also anything that is equipped with identification (tag reading), sensing or actuation capabilities, which excludes entities in the Internet domain (hosts, terminals, routers, among others). The things should also have communication, networking and processing functionalities varying according to the systems requirements”.

The results of the literature review - especially the statement that despite the large amount and diversity of devices, they play finite behaviors of identification, sensing or acting – have helped us observing the existence of recurrent interaction flows among IoT elements. From this conjecture, we led a reasoning process on the three behaviors and the IoT properties in order to ground the composition of IoT Interaction Arrangements.

The set of nine IIAs grounded the proposal of a software technology that meets our research problem. We constructed an approach called ScenarI$_{oT,}$ which provides information Catalogs to support the description of scenarios in the development of IoT-bSS.

Two assessment studies have been performed in two distinct undergrad projects. The results show that the $ScenarI_{oT}$ was useful to describe scenarios and therefore to complement systems requirements specification.

Considering the results obtained and the analyses conducted, we can answer the main research question of this work arguing that scenario specification of IoT-bSS can be performed with the support of the $ScenarI_{oT}$ approach, as it was development grounded on IIAs that capture entities and their interaction in the IoT realm and as the approach are based on Catalogs that capture concrete information to be captured in the scenarios descriptions.

The remaining of this chapter describes the resulting contributions, limitations, and insights for future works.

7.2 Contributions

The main contributions of this work revolve around the following:

- Organization of the *state-of-the-art* of the IoT area. IoT concepts and properties were analyzed and organized through an StLR. Although we found works surveying the technical literature on IoT, those do not provide a structured methodology and reporting. We conducted the StLR so that it could be replicated and the analysis could be ground on relevant data. From the concepts and properties we could propose a software technology, and you hope those results can support future works.
- A set of IIAs was developed which have been recognized its utility for the description of scenarios, architecture definition, and documentation, and features idealization. This set of IIAs can be harnessed for further research and development activities from practitioners.
- It provides an approach that aims at supporting describing scenarios in projects with an IoT background. The manner and resources by which this approach was developed can inspire the development of other software technologies considering the properties of the IoT realm.

7.3 Limitations

The identified boundaries of this work are:

- On the StLR that was conducted we highlight some threats such as (a) the fact only Scopus was used as a search engine, which can imply in missing some important works; (b) inconsistent terminology and restrictive keywords. In order to reduce the researchers' bias, it has been searched for other reviews and observed the adopted terms to compose our search string; (c) data extraction and interpretation biases; (d) the quality assessment regarding the research methodology of studies could not be performed.

- The processes of IIAs development and composition of the Catalogs were conducted in an *ad-hoc* manner. Therefore, the resulting assets of this work carry the biases related to the conceptual background of this dissertation's author.
- There were no empirically experimented assessments. Observational studies were carried out with students, not practitioners from the industry or IoT specialists.

7.4 Future Work

The possibility of future work are as follows:

- More in-depth analysis can be conducted on the usefulness of the ScenarI$_{oT}$ approach and the resources we have developed - including the IIAs. These assets can be applied in real projects so that a better assessment can be made.
- Evolution of the ScenarI$_{oT}$ approach considering the suggestions reported from the studies carried out in this dissertation and also further assessment studies in the industry.
- The IIAs can be explored and tailored in a lower abstract level they were developed, so that they can capture other concrete entities and responsibilities such as data storage, load balancing, etc., which are pertinent to later phases of development such as architecture definition, structural systems design, etc.
- Further research can be conducted considering the statement that scenario descriptions artifacts can be clustered within use cases artifacts.
- The IIAs can be seen as a means to enable reusing systems requirements and other system artifacts if established and maintained some association with IIAs as proposed with the scenarios.
- A tool can be developed in order to automate the process of identifying IIAs for beginners in the IoT realm.
- The three behaviors (Identification, Sensing and Acting) have been guiding all through this work, and they can be more explored in research on composing, developing or engineering IoT systems, at least in the functional

perspective, as the three behaviors are high contrasting against our traditional software-related thinking mode.

7.5 Dissertation conclusions

This last chapter presented a summary of the activities performed, the contributions and limitations of this work, besides reporting the topics that might be researched in the future. As discussed in the early chapters of this dissertation, the IoT field is very young. This new paradigm brings challenges for many research fields, and the Software Engineering area is not excluded. IoT can bring challenges from early to later phases of development, and we have focused on the challenges related to activities of specifying systems scenarios, giving a step further research combining IoT and SSE. We hope that the results of this work and the possibilities of future work can contribute to the progress of research on IoT from the lens of the Software Engineering research field.

References

[1] AHMED, A. H.; OMAR, N. M.; IBRAHIM, H. M. Modern IoT Architectures Review: A Security Perspective. In **8th Annual International Conference on ICT: Big Data, Cloud and Security**, 2017.

[2] AL-FUQAHA, A. *et al.* Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. In **IEEE communications surveys & tutorials**, v. 17, n. 4, p. 2347-2376, 2015.

[3] ALEXANDER, I.; MAIDEN, N.; Scenarios, stories, and use cases: the modern basis for system development. In **Computing and Control Engineering**, v. 15, n. 5, p. 24-29, 2004.

[4] DE ALMEIDA, B. J. C. *et al.* Scientific research ontology to support systematic review in software engineering. *In* **Advanced Engineering Informatics**, v. 21, n. 2, p. 133-151, 2007.

[5] ALQASSEM, I.; SVETINOVIC, D. A taxonomy of security and privacy requirements for the Internet of Things (IoT). In: **2014 IEEE International Conference on Industrial Engineering and Engineering Management**. IEEE, p. 1244-1248, 2014.

[6] ALQASSEM, I. Privacy and Security Requirements Framework for the Internet of Things (IoT). In ***Companion Proceedings of the 36th International Conference on Software Engineering***, 739–741, 2014.

[7] ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. **Comput. Networks***, v.* 54, n. 15, p. 2787–2805, 2010.

[8] BANDYOPADHYAY, D.; SEM, J. Internet of things: Applications and challenges in technology and standardization. In **Wireless personal communications**, v. 58, n. 1, p. 49-69, 2011.

[9] BAO, Y.; QU, Q.; YAO, Y. An object-oriented XML description specification for scenario entity. In ***ACM International Conference Proceeding Series***, p. 25–29, 2017.

[10] CALDIERA, G; BASILI, V. R.; The Goal Question Metric Approach. In **Encyclopedia of software engineering**, p. 528-532, 1994.

[11] BASSI, A. *et al*. **Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model.** Springer Publishing Company, 2013.

[12] BEHRENS, H. Requirements analysis using statecharts and generated scenarios. In **Doctoral Symposium at IEEE Joint Conference on Requirements Engineering**. 2002.

[13] BELL, T. E.; THAYER, T. A. Software Requirements: Are They Really a Problem?

In **Proceedings of the 2Nd International Conference on Software Engineering,** p. 61–68, 1976.

[14] BORGIA, E. The internet of things vision: Key features, applications and open issues. **Computer Communications**, v. 54, p. 1-31, 2014.

[15] BORTENSCHLAGER, M. *et al.* A context-sensitive infrastructure for coordinating agents in ubiquitous environments. In **Multiagent Grid Syst**.*v.* 5, n. 1, p. 1–18, 2019.

[16] BROCK, D. L. The Electronic Product Code (EPC) – A Naming Scheme for Physical Objects. **Auto-ID Cent. White Pap. MIT-AUTOID-WH-002**, 2001.

[17] BROCK, D. L. Integrating the Electronic Product Code (EPC) and the Global Trade Item Number (GTIN). **MIT Auto-Id Cent.** 1–25, 2002.

[18] BROOKS, F. P. No Silver Bullet Essence and Accidents of Software Engineering. In **IEEE computer**, v. 20, n. 4, p. 10-19, 1987.

[19] BURG, J. F. M.; VAN DE RIET, R. P. A Natural Language and Scenario based Approach to Requirements Engineering. In **Proceedings of Workshop in Natuerlichsprachlicher Entwurf von Informationssystemen,** 1996.

[20] CARROLL, J. M. Becoming social: Expanding scenario-based approaches in HCI. In **Behaviour & Information Technology**, v. 15, n. 4, p. 266-275, 1996.

[21] CAVALCANTE, E. *et al.* An analysis of reference architectures for the internet of things. In **1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures (CobRA)**, 1–4, 2015.

[22] CHADDA, A.; THIGALE, M.; BRITTO, P. Medical Device to Tackle Neurogenic Bladder. In **7th WACBE World Congress on Bioengineering 2015**. Springer, Cham, 2015. p. 178-181.

[23] COCKBURN, A. Using Goal-Based Use Cases. **Journal of object-oriented programming**, v. 10, n. 7, p. 56-&, 1997.

[24] CONSEL, C.; KABÁČ, M. Internet of Things: From Small to Large-Scale Orchestration. In **IEEE 37th International Conference on Distributed Computing Systems (ICDCS)**, 1748–1755, 2017.

[25] ABOWD, G. D., *et al.* Towards a Better Understanding of Context and Context-Awareness. In: **International symposium on handheld and ubiquitous computing**. Springer, Berlin, Heidelberg, p. 304-307, 1999.

[26] DUNKELS, A.; VASSEUR, JP. *The Internet of Things : IP for Smart Objects.* **White Paper**, v. 1, 2008.

[27] DYBA, T.; DINGSOYR, T.; HANSSEN, G. K. Applying Systematic Reviews to Diverse Study Types: An Experience Report. In **First International Symposium on Empirical Software Engineering and Measurement,** 225–234, 2007.

[28] FALUDI, R. *et al.* Botanicalls, 2001. Available in : <*https://www.botanicalls.com/*>. Accessed in 2019 April.

[29] FERNÁNDEZ, D. M. Supporting Requirements-Engineering Research That Industry Needs: The NaPiRE Initiative. **IEEE Software**, v. 35, n. 1, p. 112-116, 2018.

[30] FLOERKEMEIER, C.; *et al.* **The Internet of Things: First International Conference, IOT 2008, Zurich, Switzerland, March 26-28, 2008**.

[31] FREMANTLE, P. A reference architecture for the internet of things. **WSO2 White paper**, 2015.

[32] GIL, D. *et al.* Internet of Things: A Review of Surveys Based on Context Aware Intelligent Services. ***Sensors***, v. 16, n. 7, 1069.

[33] GLINZ, M. Improving the quality of requirements with scenarios. In **Proceedings of the second world congress on software quality**, 55–60, 2000.

[34] GLINZ, M. Rethinking the Notion of Non-Functional Requirements. In **Proceedings of the Third World Congress for Software Quality***,* 55–64, 2005.

[35] GO, K.; Carroll, J. M. The Blind Men and the Elephant: Views of Scenario-based System Design. **Interactions,** v. 11, n. 6, p. 44–53, 2004.

[36] GUBBI, J. *et al.* Internet of Things (IoT): A vision, architectural elements, and future directions. **Futur. Gener. Comput. Syst.** v. 29, n. 7, p. 1645–1660. 2013.

[37] Guo, B. *et al.* Opportunistic IoT: Exploring the social side of the internet of things. In: **Proceedings of the IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)**, p. 925-929, 2012.

[38] GUSMEROLI, S.; SUNDMAEKER, H.; BASSI, A. Internet of Things Strategic Research Roadmap. **Clust. Eur. Res. Proj. Tech. Rep**, 9–52, 2015.

[39] HAUMER, P. *et al.* Bridging the gap between past and future in RE: a scenario-based approach. In **Proceedings IEEE International Symposium on Requirements Engineering**, 66–73, 1999.

[40] HOLLER, J. *et al.* ***From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence****.* 1st ed. London, United Kingdom: Academic Press Ltd, 10 Apr 2014.

[41] HONG, R.; XIONG, Z.; ZHOU., Q. A simulating method of dynamic inter-connection between models. In **AIAA Modeling and Simulation Technologies Conference***, 2016*.

[42] HU, Y. The Plan of Elderly Smart Community: Based on the Concept of Internet of Things. In **Management and Service Science (MASS), International Conference**, 1–4, 2010.

[43] HULL, E.; JACKSON, K.; DICK, J. ***Requirements Engineering*** **(3rd ed.)**.

Springer-Verlag, Berlin, Heidelberg, 2017.

[44] JOSHI, Y. V. Information visibility and its effect on supply chain dynamics. Massachusetts Institute of Technology (Doctoral dissertation, Massachusetts Institute of Technology), 2000.

[45] KARAGIANNIS, C. *et al.* A survey on application layer protocols for the internet of things. **Trans. IoT Cloud Comput.** v. 3, n. 1, p. 11–17, 2015.

[46] KITCHENHAM, B. A.; DYBA, T.; JORGENSEN, M. Evidence-based software engineering. In **Proceedings. 26th International Conference on Software Engineering**, 273–281, 2004.

[47] KOLOKOTSA, D. *et al.* Development of an intelligent indoor environment and energy management system for greenhouses. **Energy Convers. Manag**. 51, 1, 155–168, 2010.

[48] LALECI, G. B. *et al.* A platform for agent behavior design and multi agent orchestration. In **Lecture Notes in Computer Science**, 205–220, 2005.

[49] LARRUCEA, X. *et al.* Software Engineering for the Internet of Things. ***IEEE Software***. 34, 1, 24–28, 2017.

[50] LEE, B.; MILLER, J. Multi-project management in software engineering using simulation modelling. **Software Quality Journal.** 12, 1, 59–82, 2004.

[51] LI, S. *et al.* The internet of things: a survey. *in* **Information Systems Frontiers**, v. 17, n. 2, p. 243-259, 2015.

[52] LIU, L.; YU, E. K. Designing information systems in social context: a goal and scenario modelling approach. **Information systems**, v. 29, n. 2, p. 187-203, 2004.

[53] LOMBRISER, P. Engaging Stakeholders in Scenario-Based Requirements Engineering with Gamification, 2015 (*Master's thesis*).

[54] MADAKAM, S.; RAMASWAMY, R.; TRIPATHI, S. Internet of Things (IoT): A Literature Review. **Journal of Computer and Communications**, v. 3, n. 05, p. 164, 2015.

[55] MAIDEN, N. M. CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements. In: **Domain Modelling for Interactive Systems Design**. Springer, Boston, MA, 1998. p. 39-66, 1998.

[56] MATALONGA, S.; RODRIGUES, F.; TRAVASSOS, G. 2015. Challenges in testing context aware software systems. In: **9th Workshop on Systematic and Automated Software Testing**. p. 51-60, 2015.

[57] MIORANDI, D. *et al.* Internet of things: Vision, applications and research challenges. **Ad Hoc Networks,** v. 10, n. 7, p. 1497–1516, 2012.

[58] MORIN, B.; HARRAND, N; FLEUREY, F. Model-Based Software Engineering to Tame the IoT Jungle. **IEEE Software**, v. 34, n. 1, p. 30-36, 2017.

[59] MOSHIRPOUR, M. *et al.* Model based approach to detect emergent behavior in multi-agent systems. In **12th International Conference on Autonomous Agents and Multiagent Systems***,* 1285–1286. *2013.*

[60] MOTTA, R. C.; OLIVEIRA, K. M. D.; TRAVASSOS, G. H. Rethinking Interoperability in Contemporary Software Systems. In **2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS),** 9–15, 2017.

[61] MOULIN, B.; BRASSARD, M. A scenario-based design method and an environment for the development of multiagent systems. In: **Australian Workshop on Distributed Artificial Intelligence**. Springer, Berlin, Heidelberg, 1995. p. 216-232.

[62] NARDI, B. A. **The Use of Scenarios in Design**. Hewlett-Packard Laboratories, Technical Publications Department, 1992.

[63] NUSEIBEH, B.; EASTERBROOK, S. In: **Proceedings of the Conference on the Future of Software Engineering**. ACM, 2000. p. 35-46.

[64] PARIDEL, K. *et al.* Middleware for the internet of things, design goals and challenges. **Electronic Communications of the EASST**, v. 28, 2010.

[65] POTTS, C.; TAKAHASHI, K.; ANTON, A. I. Inquiry-based requirements analysis. **IEEE software**, v. 11, n. 2, p. 21-32, 1994.

[66] LEITE, J. C. S. do P. A Scenario Construction Process. **Requirements Engineering**, v. 5, n. 1, p. 38-61, 2000.

[67] RALYTÉ, J. Reusing Scenario Based Approaches in Requirement Engineering Methods: CREWS Method Base. In **Proceedings of the 10th International Workshop on Database & Expert Systems Applications,** 305-309, 1999.

[68] RALYTÉ, J.; ROLLAND, C.; PLIHON, V. Method Enhancement with Scenario Based Techniques. In **Advanced Information Systems Engineering**, 103–118, 1999.

[69] REGNELL, B.; RUNESON, P. Combining Scenario-based Requirements with Static Verification and Dynamic Testing. In **Proc. Fourth Intern. Workshop on Requirements Engineering: Foundation for Software Quality***,* 1998.

[70] REN, S.; RUI, K.; BUTLER, S. Refactoring the Scenario Specification: A Message Sequence Chart Approach. In **Object-Oriented Information Systems,** 294–298, 2003.

[71] ROLLAND, C. *et al.* A proposal for a scenario classification framework. **Requirements Engineering**, v. 3, n. 1, p. 23-47, 1998.

[72] ROLLAND, C.; ACHOUR, C. B. Guiding the construction of textual use case

specifications. **Data & Knowledge Engineering**. 25, 1, 125–160. 1998.

[73] SCHNEIDER, S. *et al.* Prioritisation of simulation models for ensuring safety and security in underground stations on the basis of a detailed requirements analysis. In: **Future Security Research Conference**. Springer, Berlin, Heidelberg, 2012. p. 117-128.

[74] SETHI, P.; SARANGI, S. R. Internet of Things: Architectures, Protocols, and Applications. **Journal of Electrical and Computer Engineering**, 2017.

[75] SIKORA, E.; DAUN, M.; POHL, K. Supporting the Consistent Specification of Scenarios across Multiple Abstraction Levels. In: **International Working Conference on Requirements Engineering: Foundation for Software Quality**. Springer, Berlin, Heidelberg, 2010. p. 45-59.

[76] SINGH, D.; TRIPATHI, G.; JARA, A. J. A survey of Internet-of-Things: Future vision, architecture, challenges and services. **IEEE World Forum Internet Things, WF-IoT**, 2014.

[77] SPINELLIS, D. Software-Engineering the Internet of Things. **IEEE Software**, v. 34, n. 1, p. 4-6, 2017.

[78] SUNDMAEKER, H. *et al.* Vision and challenges for realising the Internet of Things. **Cluster of European Research Projects on the Internet of Things, European Commision**, v. 3, n. 3, p. 34-36, 2010.

[79] SUTCLIFFE, A.; THEW, S.; JARVIS, P. Experience with user-centred requirements engineering. **Requirements Engineering**, v. 16, n. 4, p. 267-280, 2011.

[80] SUTCLIFFE, A. Scenario-based requirements analysis. **Requirements Engineering**. 3, 1, 1998.

[81] TAIVALSAARI, A.; MIKKONEN, T. A Roadmap to the Programmable World: Software Challenges in the IoT Era. **IEEE Software**. 34, 1, 2017.

[82] TRAPPEY, A. J. C. *et al.* A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. **Advanced Engineering Informatics**, v. 33, p. 208-229, 2017.

[83] UCKELMANN, D.; HARRISON, M.; MICHAHELLES, F. An Architectural Approach Towards the Future Internet of Things. In: **Architecting the internet of things**. Springer, Berlin, Heidelberg, 2011. p. 1-24.

[84] VAT, K. H. Teaching HCI with scenario-based design: The constructivist's synthesis. In **Proceedings of the Conference on Integrating Technology into Computer Science Education**, 9–12, 2001.

[85] VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Modelos e processos para o desenvolvimento de sistemas sensíveis ao contexto. **André Ponce de Leon F.**

de Carvalho, Tomasz Kowaltowski.(Org.). Jornadas de Atualização em Informática, p. 381-431, 2009.

[86] VILAIN, P.; SCHWABE, D.; DE SOUZA, C. S. **Use Cases and Scenarios in the Conceptual Design of Web Applications**. PUC, 2000.

[87] WEWERS, M. E.; LOWE, N. K. A critical review of visual analogue scales in the measurement of clinical phenomena. **Research in nursing & health**, v. 13, n. 4, p. 227-236, 1990.

[88] WHITMORE, A.; AGARWAL, A.; DA XU, Li. The Internet of Things-A survey of topics and trends. **Information Systems Frontiers**, v. 17, n. 2, p. 261-274, 2015.

[89] WHITTLE, J.; KRÜGER, I. A methodology for scenario-based requirements capture. In **Proceedings of the ICSE 2004 Workshop on Scenarios and State Machines***,* 2004*.*

[90] WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: **Proceedings of the 18th international conference on evaluation and assessment in software engineering**. ACM, 2014. p. 38.

[91] XIAOJIANG, X.; JIANLI, W.; MINGDONG, L. Services and key technologies of the internet of things. **ZTE Communications**, v. 2, p. 011, 2010.

[92] ZAMBONELLI, F. Key Abstractions for IoT-Oriented Software Engineering. **IEEE Software**, v. 34, n. 1, p. 38-45, 2017.

[93] ZAMBONELLI, F. Towards a general software engineering methodology for the Internet of Things. **arXiv preprint arXiv:1601.05569**, 2016.

[94] ZHU, H. Formal reasoning about emergent behaviours of multi-agent systems. In **17th International Conference on Software Engineering and Knowledge Engineering, SEKE,** 280–285, 2005.

[95] ZHU, H.; JIN, L. Scenario analysis in an automated tool for requirements engineering. **Requirements Engineering**, v. 5, n. 1, p. 2-22, 2000.

[96] IEEE Standard for Application and Management of the Systems Engineering Process. *IEEE Std 1220-1998*, 1999.

Appendix

Annexes are available in: *https://is.gd/5YoXdJ*