ScenarI$_{oT:}$ Support for Scenario Specification of Internet of Things-based Software Systems

Valéria M. da Silva[1], Guilherme H. Travassos[1]

[1]Department of Systems Engineering and Computer Science – Federal University of Rio de Janeiro (COPPE/UFRJ).
Rio de Janeiro - RJ, Brazil.

valeriams.dev@gmail.com, ght@cos.ufrj.br

Abstract. *As well as any paradigm shift, the Internet of Things (IoT) brings up challenges related to several areas of research, including Software Engineering at different stages of development. This work proposes the ScenarI$_{oT}$ approach aiming to support the specification of scenarios when developing software systems on the IoT background. This approach is grounded on Interaction Arrangements - designed by applying an evidence-based approach – which represents recurrent flows of interaction between abstract elements of the IoT domain. These results presented can contribute to the progress of research on IoT from the lens of the Software Engineering research field.*

Resumo. *Assim como qualquer mudança de paradigma, a Internet das Coisas (IoT) traz desafios relacionados a diversas áreas de pesquisa, incluindo Engenharia de Software em diferentes estágios de desenvolvimento. Este trabalho propõe a abordagem ScenarIoT com o objetivo de apoiar a especificação de cenários no desenvolvimento de sistemas de software no contexto de IoT. Essa abordagem é baseada em arranjos de interação - projetados aplicando-se uma abordagem baseada em evidências - que representam fluxos recorrentes de interação entre elementos abstratos do domínio IoT. Os resultados apresentados podem contribuir para o avanço das pesquisas em IoT sob a ótica da área de pesquisa em Engenharia de Software.*

1. Introduction

There will come a day when computers would have to gather information from things that could “see,” “hear,” “smell,” and “touch” the world from themselves. It means 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: the IoT.

Every paradigm changing can impact and bring new challenges for the involved field. According to Li, Xu, and Zhao (2015), “*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).

Research combining IoT and SSE is ongoing. Around the years 2016 and 2017, some authors (Larrucea et al. 2017; Spinellis 2017; Taivalsaari and Mikkonen 2017; Franco Zambonelli 2016; F Zambonelli 2017; Da Silva et al. 2017) have pointed out roadmaps and also possible software challenges to face when developing IoT-based Software Systems (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*" (Larrucea et al. 2017). Such context reflects that there are many challenges behind engineering IoT-bSS – many still unaware – besides the lack of answers, but the first steps have been arising toward addressing the known issues so far.

Therefore, this work attempts at taking another step concerned with the early phases of IoT-bSS development, focusing on the capture and specification of the 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 carried out at the Experimental Software Engineering (ESE) group at COPPE/UFRJ, each project with its peculiarities.

Therefore, three issues have been used as drivers for the present research, 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; this knowledge is 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 without enough knowledge of IoT and its capabilities.

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 (Ren Shengbing and Rui 2003; Sikora Ernst and Daun 2010; Sutcliffe 1998). The proposal of using or tailoring some existing scenario-based approach 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*" (Burg and van de Riet 1996). 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. Thus, the following research question was formulated:

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

This work is organized as follows, which reflects the conducted research methodology. The IoT concepts are described in the next section; section 3 presents the design of IoT interaction arrangements, which are grounded on the results of the literature review. Section 4 presents a proposal for a software technology based on the arrangements described previously, followed by the presentation of assessment studies and the conclusion in section 6.

2. Internet of Things

As mentioned before, 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. For this purpose, we undertook a Structured Literature Review (SLR) aiming at gaining an in-depth understanding of the IoT research domain and an overview of areas that have been applying it. The review protocol was performed following recommendations proposed by (Kitchenham, Dyba, and Jorgensen 2004). The Scopus[1] engine was chosen for this review as it indexes several databases of peer-reviewed sources (Matalonga, Rodrigues, and Travassos 2015).

The full review protocol and the results in details are shared by Motta, Silva, and Travassos (2019). The main questions and answers of the SLR to bring the resulting IoT concepts to this work were presented as follows:

- (RQ1) What is the “Internet of Things”?
- (RQ2) Which characteristics can define an IoT domain?
- (RQ3) Which are the areas of IoT application?

(RQ1): 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.

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.

(RQ2): The 15 papers provided 263 excerpts in the process of textual analysis, from what we could identify 21 characteristics: Accuracy, Adaptability, Availability, Connectivity, Context-Awareness, Efficiency, Extensibility, Flexibility, Manageability, Modularity, Performance, Privacy, Reliability, Robustness, Scalability, Smartness, Sustainability, Traceability, Trust, Ubiquity, Visibility.

(RQ3): Several application domains will leverage the Internet of Things paradigm advantages. “The domain of the application areas for the IoT is limited only by imagination” (Whitmore, Agarwal, and Da Xu 2015). We attempted to categorize some of the application areas from the domains of Mobility, Healthcare, Security, among

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

others, into the tree behaviors - identification, sensing, and actuation - as presented in (Motta, Silva, and Travassos 2019).

3. IoT Interaction Arrangements

It has been observed actual flows when reasoning about the IoT concept and the set of three behaviors. As previously discussed, the Internet of Things "is a paradigm that enables the composition of systems from various devices." Thus, the composition opportunities of applications 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, we can interpret that despite the large amount and diversity of devices, they play finite behaviors of identification, sensing, or acting. This statement is an essential driver to this research because the three IoT behaviors are high contrasting against the traditional software-related mindset and thus relevant to be considered when composing, developing, or engineering IoT systems.

The composition of IoT-bSS requires the entities to be orchestrated to turn received data into actionable information (Consel and Kabáč 2017) and to support a business process workflow (Holler et al. 2014). 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, things depend on other things to trigger their capabilities, either from an actor direct intervention (system or individual[2]) 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 mapped 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[3]:

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

Scenario description: Heartbeat data is collected from one patient; the data is shared, and it can 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, and it can be accessed from displays (devices); some worker visualizes data from his/her display.

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

[2] Living beings, mainly humans.
[3] In the context of this work we are addressing scenarios as: "[Scenario] is an ordered set of interactions between partners, usually between a system and a set of actors external to the system" (Glinz 2000).

The possibility of knowing a set of recurrently implemented IoT interaction flows has raised a conjecture that those flows 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 conjecture, 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. The outputs of each step have been arranged into structures called **IoT Interaction Arrangements** (IIAs), representing interaction flows among things and other non-IoT elements, regardless of domain.

Two main steps have been taken toward structuring and designing IIAs. They are two **reasoning rounds** being grounded on the results of the literature review, where the three behaviors (Identification, Sensing, and Actuation) were input to the first step and the 21 IoT properties (e.g., Adaptability, Autonomy, Context Awareness) to the second one. The next sub-sections address the execution of this reasoning process, presenting the resulting IIAs from each step as well as application examples to illustrate the instantiation of each arrangement.

3.1. Reasoning on the three IoT Behaviors

In this first step, it 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).

We used two classes - Data Producers and Actuators – according to the behaviors that things play in order to guide the reasoning in this first step.

3.1.1 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 Producers** (DPs). It means that, in the moment of arranging the IoT interactions, things performing behaviors of Identification and Sensing are being treated as DPs

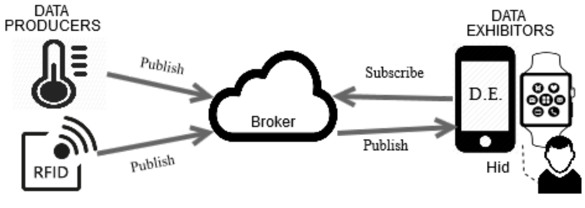

Figure 1 - IIA-1: IoT Data Exhibition

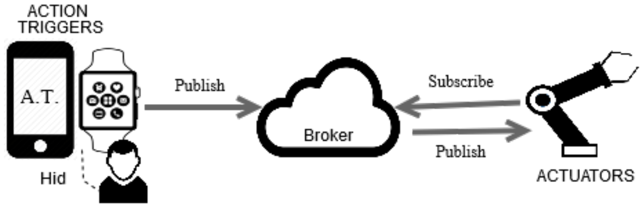

Figure 2 - IIA-2: Actuation triggered by an individual

equivalently, mainly because they play the same higher-level task and thus interact similarly with other components.

Those elements that consume data for exhibition purposes have been called **Data Exhibitors** (DEs). It means devices that enable data visualization by individuals who are candidates to interact with DPs. Therefore, considering the interaction among these two elements - DPs and DEs – with the purpose of "Data Exhibition," the first IIA has been designed, the so-called IIA-1[4] (Figure 1). This first and most basic interaction arrangement relies on data collection from DPs (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).

A supply chain management software can be considered an actual instantiation example of the IIA-1, as the primary objective is to enable object tracking in the supply chain for companies, by reading and sharing data from object's tags so that they can be read/visualized by stakeholders from their HIDs (Joshi 2000). The systems of the previously described scenarios for temperature and heartbeat monitoring are also examples of IIA-1 instantiation.

3.1.2 Actuators

Actuators can receive commands from human actors by his/her HIDs, leading to the design of IIA-2 in which actuation is triggered by individuals (Figure 2). 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, it 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 (F Zambonelli 2017).

Software system elements can interact with actuators, composing the IIA-3 (Figure 3). 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, in this case, 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

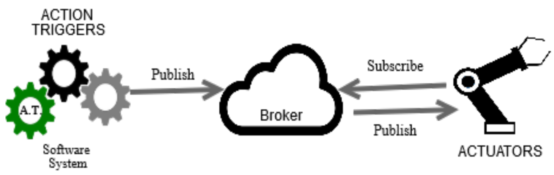

Figure 3 - IIA-3: Actuation triggered by a software system

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

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 executed by IoT actuators. Thus, this flow of visualizing IoT data, making decisions, and manually triggering the actuator's actions have led us to the design of IIA-4 (Figure 4). This interaction arrangement represents functional units where IoT data support individuals on decisions of triggering actions. A system to support *urinary bladder* functions (Chadda, Thigale, and Britto 2015) can be assumed to exemplify the 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.

3.2. Reasoning on the 21 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 behaves. For instance, considering the "Context-Awareness" property, we focused on how it can influence systems in a functional sense. It resulted in an argument that Context-Awareness concerns of capturing context information and reacting according to information collected. This functional trait has led to the identification of interaction flows among elements with the following responsibilities: data collection, decision-making, automatic action triggering, and actuation performing.

The IIA-5 (Figure 5) represents functional units for "Actuation triggered by software systems based on IoT data" and brings together IoT elements on both sides. That is, sensors or tag readers are DPs; 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 on decisions, and the system reacts according to information collected. Kolokotsa *et al.* (2010) present an

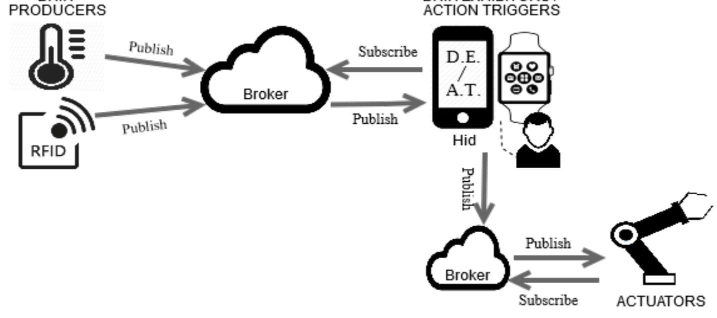

Figure 4 - IIA-4: Actuation triggered by an individual, based on IoT data.

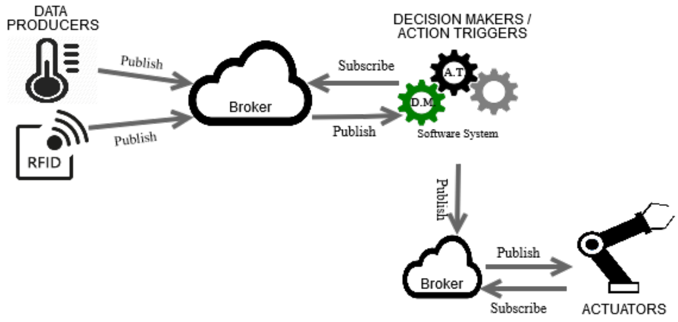

Figure 5 - IIA-5: Actuation triggered by a software system, based on IoT data.

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; 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.

That reasoning procedure has been performed for each IoT property, resulting in the following IIAs (Figure 6, Figure 7) and IIA-8 / IIA-9 presented by Da Silva 2019. IIA-6 can be instantiated in an assumed system for automated doors locking of data centers. DPs 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.

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. The instantiation example of this arrangement can be the Twitter-enabled sensing system called *Botanicalls*[5]. It aims to keep individuals informed about the health state of their plants.

As an example of IIAs applicability on SSE, the following works have already applied it in the context of software inspection and to analyze IoT properties deeply (De Souza et al. 2019; De Souza et al. 2020a; De Souza et al. 2020b).

4. Towards an IoT Scenario Description Approach

Once a more in-depth understanding of the IoT realm has been obtained from the literature review, nine IIAs could emerge by observing current flows among IoT elements. The set of raised IIAs supports conjecturing that, in general, scenario descriptions can represent instantiations of one IIA or a combination of specific ones.

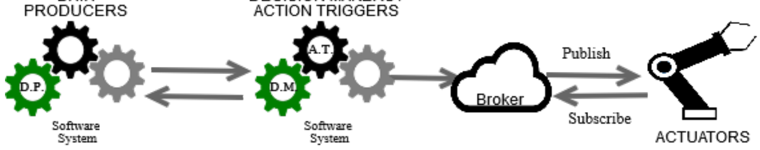

Figure 6 - IIA-6: Actuation triggered by a software system, based on non-IoT data.

[5] www.botanicalls.com accessed in June 2020.

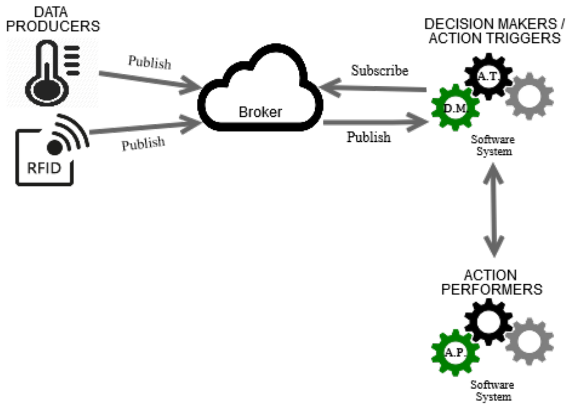

Figure 7 - IIA-7: Non-IoT actuation triggered by a software system, based on IoT data.

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.

The process of extracting information from each element involved 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.

After extracting the inquiries/information from each abstract entity existing in the IIAs, they were organized according to each arrangement and compiled into nine Catalogs[6]. 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.

These information catalogs compose the proposed **ScenarI$_{oT}$** software approach, which process for achieving this technology started in summary from IoT concepts and its behaviors, mapping IIAs, and extracting relevant inquiries into nine catalogs to be used as a basis to assist writing scenarios descriptions.

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 corresponding Catalog and writing 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

[6] The list of all nine information Catalogs for each IIA can be found in the link: https://bityli.com/KGIPH

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. Considering that one may encounter difficulties in the task of

2) identifying a related IIA, a decision-making flowchart has been designed to provide a guide for that task. The flowchart is available in this link[7].

3) ***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.

For instance, considering our heart-monitoring scenario, the IIA-1 has been chosen in the first step. Its corresponding 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 1, 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.

The resulting scenario description was written in natural language, as shown in

. 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 prerequisite for the **ScenarI$_{\mathbf{oT}}$** approach: the analysts/practitioners in charge of describing the scenarios by employing the ScenarI$_{oT}$ need to have previous knowledge on scenario framing.

Table 1 – IIA-1 Information Catalog

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

[7] The decision-making flowchart can be found in the link: https://bityli.com/KGIPH

Table 2 – Scenario description

Heart monitoring scenario description
In some hospitals, 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. Assessment studies

The $ScenarI_{oT}$ approach proposed in this work was employed in two projects, part of distinct undergrad courses and at different universities. 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.

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 respective project. The studies and results are briefly described in the next subsections.

5.1. Assessment study I: "Camarão Iotizado" Project

This study is an initial evaluation of the $ScenarI_{oT}$ approach from 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 Catalog of information.

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

All the students enrolled in the undergrad course participated in the study. They were undergraduate students from different fields, Computer Engineering and Electronic Engineering. Students had no knowledge of IoT concepts but a vague reference on the term.

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

The participants described some scenarios; see an example in Table 3. The participants employed the Guide to support them in identifying the pertinent IIAs. Just for a matter of facilitating the reading and analysis of information on the artifact, Data Producers are highlighted in green, Decision makers/Action triggers highlighted in blue, and Action performers in yellow.

Table 3 - Scenario description

(IIA-7) Scenario: The manager receives the 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 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.

Considering the points that have been observed in this study, we can answer the main research question 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 project the system. As participants reported, the scenarios artifacts were blended with use cases description artifact and therefore used in conjunction.

5.2. Assessment study II

This study was carried out in a *Programming Topics* undergrad course at a different university in 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.

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. For this reason, the main question was:

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

The students performed brainstorming activities to idealize and elect the systems' features before the study application. 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 the concept of scenario description, and also to present IoT concepts and the elements that compose the ScenarI$_{oT}$ 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 ScenarI$_{oT}$ approach and 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. Besides associating IIAs, some groups also incremented/tailored the original representations with more accurate information. An example is the one depicted in Figure 8.

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. The students already had 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.

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

5.3. Results and Discussion

From the results obtained, we could conclude that the ScenarI_{oT} approach was useful when applying to the Camarão Iotizado project. In study II 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 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 that students acquired a more in-depth understanding of the IoT paradigm, improving their understanding of the possibilities of interactions among the elements in this realm.

6. Conclusion

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 in 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. 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 to ground the

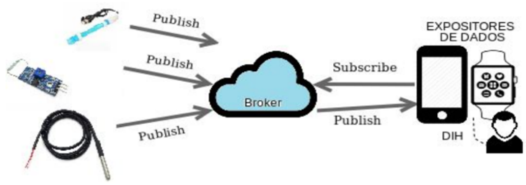

Figure 8 - Tailored IIA-1

composition of IoT Interaction Arrangements. The set of nine IIAs grounded the proposal of a software approach 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.

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 detailed information to be captured in the scenarios descriptions.

References

Burg, J F M, and R P van de Riet. 1996. “A Natural Language and Scenario-Based Approach to Requirements Engineering.”

Chadda, Akriti, Manish Thigale, and Preethika Britto. 2015. “Medical Device to Tackle Neurogenic Bladder.” In 7th WACBE World Congress on Bioengineering 2015.

Consel, C, and M Kabáč. 2017. “Internet of Things: From Small- to Large-Scale Orchestration.” In 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 1748–55. https://doi.org/10.1109/ICDCS.2017.314.

Da Silva, Valéria Martins, Schneider, Daniel Serrão, Oliveira, Luíz Felipe Silva, Souza, Jano. 2017. "Towards a framework for supporting community modeling in crowdsourcing systems". In: IEEE 21th International Conference on Computer Supported Cooperative Work in Design (CSCWD).

Da Silva, Valéria Martins. “SCENARIoT Support for Scenario Specification of Internet of ThingsBased Software Systems,” Dissertation, Federal University of Rio de Janeiro, Rio de Janeiro, 2019.

De Souza, Bruno Pedraça; Motta, Rebeca Campos; Costa, Daniella de Oliveira, Travassos, Guilherme Horta. 2019. "An IoT-based Scenario Description Inspection Technique." In Proceedings of the XVIII Brazilian Symposium on Software Quality (SBQS'19).

De Souza, Bruno Pedraça; Motta, Rebeca Campos; Travassos, Guilherme Horta. 2020a. "The first version of SCENARIotCHECK: A Checklist for IoT based Scenarios." In: Simpósio Brasileiro de Engenharia de Software (SBES), 33.

De Souza, Bruno Pedraça; Motta, Rebeca Campos; Travassos, Guilherme Horta. 2020b. "Towards the Description and Representation of Smartness in IoT Scenarios Specification." In: Insightful ideas and emerging results - Simpósio Brasileiro de Engenharia de Software (SBES). 33.

Glinz, Martin. 2000. “Improving the Quality of Requirements with Scenarios.” In *Proceedings of the Second World Congress on Software Quality*, 9:55–60.

Holler, Jan, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, and David Boyle. 2014. *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. 1st ed. Orlando, FL, USA: Academic Press, Inc.

Joshi, Yogesh Vijay. 2000. “Information Visibility and Its Effect on Supply Chain Dynamics.” Massachusetts Institute of Technology.

Kitchenham, B A, T Dyba, and M Jorgensen. 2004. “Evidence-Based Software Engineering.” In *Proceedings. 26th International Conference on Software Engineering*, 273–81.

Kolokotsa, Denia, Giorgos Saridakis, Konstantinos Dalamagkidis, S Dolianitis, and I Kaliakatsos. 2010. “Development of an Intelligent Indoor Environment and Energy Management System for Greenhouses.” *Energy Conversion and Management* 51 (1): 155–68.

Larrucea, X, A Combelles, J Favaro, and K Taneja. 2017. “Software Engineering for the Internet of Things.” *IEEE Software* 34 (1): 24–28. https://doi.org/10.1109/MS.2017.28.

Li, Shancang, Li Da Xu, and Shanshan Zhao. 2015. “The Internet of Things: A Survey.” *Information Systems Frontiers* 17 (2): 243–59. https://doi.org/10.1007/s10796-014-9492-7.

Matalonga, Santiago, Felyppe Rodrigues, and G Travassos. 2015. “Challenges in Testing Context-Aware Software Systems.” In *9th Workshop on Systematic and Automated Software Testing*, 51–60.

Motta, Rebeca, Valéria Silva, and Guilherme Travassos. 2019. “Towards a More In-Depth Understanding of the IoT Paradigm and Its Challenges.” *Journal of Software Engineering Research and Development* 7: 3:1--3:16. https://doi.org/10.5753/jserd.2019.14.

Ren Shengbingand Rui, Kexingand Butler Greg. 2003. “Refactoring the Scenario Specification: A Message Sequence Chart Approach.” In *Object-Oriented Information Systems*, 294–98

Sikora Ernstand Daun, Marianand Pohl Klaus. 2010. “Supporting the Consistent Specification of Scenarios across Multiple Abstraction Levels.” In *Requirements Engineering: Foundation for Software Quality*, 45–59. Berlin, Heidelberg: Springer Berlin Heidelberg.

Spinellis, D., 2017. “Software-Engineering the Internet of Things.” *IEEE Software* 34 (1): 4–6. https://doi.org/10.1109/MS.2017.15.

Sutcliffe, Alistair. 1998. “Scenario-Based Requirements Analysis.” *Requirements Engineering* 3 (1): 48–65. https://doi.org/10.1007/BF02802920.

Taivalsaari, A, and T Mikkonen. 2017. “A Roadmap to the Programmable World: Software Challenges in the IoT Era.” *IEEE Software* 34 (1): 72–80. https://doi.org/10.1109/MS.2017.26.

Whitmore, Andrew, Anurag Agarwal, and Li Da Xu. 2015. “The Internet of Things—A Survey of Topics and Trends.” *Information Systems Frontiers* 17 (2): 261–74.

Zambonelli, F., 2017. “Key Abstractions for IoT-Oriented Software Engineering.” *IEEE Software* 34 (1): 38–45. https://doi.org/10.1109/MS.2017.3.

Zambonelli, Franco. 2016. “Towards a General Software Engineering Methodology for the Internet of Things.” *ArXiv Preprint ArXiv:1601.05569*.