An Architectural Design Decision Model for Resilient IoT Application

ABSTRACT

The Internet of Things is a paradigm that refers to the ubiquitous presence around us of physical objects equipped with sensing, networking and processing capabilities that allow them to cooperate with each other and with their environment to reach common goals. General objectives of IoT are improving the quality of human life and optimizing industrial processes. But any threat that affects the availability of IoT application can be crucial both financially and for the safety of the physical integrity of users. This feature calls for IoT applications that remain operational and efficiently handle threats that could occur. However, design an IoT application that can handle threats is a challenge for stakeholders due to high susceptible to threats of IoT application and lack of modelling mechanisms that contemplate resilience as first class representation. In this paper, an architectural Design Decision Model for Resilient IoT Application is presented to reduce the difficulty of stakeholders in design resilient IoT application. Our approach is illustrated and demonstrate the value through modelling of a case.

CCS CONCEPTS

• **Security and privacy** → *Domain-specific security and privacy architectures.*

KEYWORDS

Resilience, Architectural Design Decision, IoT

ACM Reference Format:

. 2019. An Architectural Design Decision Model for Resilient IoT Application. In *2019: Annual Computer Security Applications Conference (ACSAC 19), December 09–13, 2019, San Juan, PR.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Internet of Things (IoT) application is defined as a collection of automated procedures and data, integrated with heterogeneous entities (hardware, software, and personnel) that interact with each other and with their environment to reach common goals [16]. IoT application has caused a significant social and economic impact due to the application domain such as industrial, smart city, and health well-being domain [8, 37]. Thus, any threat that affects the availability of IoT application can be crucial both financially and for the safety of the physical integrity of users. In this sense, one of the most critical domains is health well-being, where failures in the monitoring of patient's vital signs can have a high impact on patient safety [14]. This feature calls for IoT applications that remain operational and efficiently handle threats that could occur [15]. However, design an IoT application that can handle threats is a challenge for stakeholders due to high susceptible to threats of IoT application and lack of modelling approach that contemplate resilience as first class representation. In the following, will be discussed this difficulties.

High susceptible to Threats of IoT application occur to several reasons. First, the fundamental characteristics of IoT make applications susceptible of failures naturally. The characteristics such as heterogeneity, interconnectivity and the merge of scale of the IoT in terms of devices creates a large surface for attacks and failures. Because systems are becoming increasingly sophisticated with arising issues of interoperability and maintenance. The complexity of heterogeneous objects would keep growing past the point of human ability to manage all smart objects [3]. Second, the devices in IoT are usually deployed in a highly dynamic, uncontrolled, sometimes remote and even hostile environments where they are connected in an unreliable way. This gives attackers a chance for physical attacks, and it is even harder to manage security in such a case [3]. Third, IoT uses wireless technology for communication and wireless communication is easier to compromise. Wireless technologies are susceptible to interference and interception as well, and a determined adversary cannot be stopped from mounting a Denial of Service attack. In general, most of the components of IoT like end devices lack high computing resources. This serves as a roadblock to the implementation of more secure and complex security protocols and become IoT components critically vulnerable to threats. Examples of threats are communication loss between

ACSAC 19, December 09–13, 2019, San Juan, PR

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

devices, process crash and unavailability of the system due to power outage, malicious programs, hacking, inadequate security policies, physical accidents, malfunction, out-dated system or software and man-in-the-middle (MITM) attack. Finally, the world of things is much more dynamic and mobile than the world of computers, with contexts changing rapidly and in unexpected ways. But we would still want to rely on things functioning properly. Structuring an Internet of Things in a robust and trustworthy manner would require redundancy on several levels and an ability to automatically adapt to changed conditions [29].

One way to minimize the problem cited above is to design an IoT application as a resilient system. A resilient system is basically a system capable of resisting various types of disturbances and recovering fully or partially [15, 33]. One IoT application containing the constraints for resilience such as redundancy, self-configure, self-heal, self-optimise and self-protect is a solution for dealing with any type of threat that may occur [27]. Resilience should be addressed in the early stages of the design phase and one way of doing this would be with Architecture design decisions (ADDs) [15, 27]. They are considered first class entities when architecting software systems. ADDs do capture potential alternative solutions, and the rationale for deciding among competing solutions [22]. But most of the models of ADDs present in the literature are generic. They do not present the necessary and specific concepts for dealing with resilience in IoT application [34]. The [15] indicates that the creation of resilience meta-models for constructing models of resilient IoT applications can help the stakeholders in design phase to create an architectural foundation. These models will be used for analyze of all possible behaviors, reduces complexity and allows a view of the system from a global point of view due to the high level of abstraction. Once behaviors are recognized, understood, and classified in a model they will be used as the insights into architecting, designing, and engineering resilient ultra-large-scale systems.

In view of the points raised above, we propose Architectural Design Decision Model for Resilient IoT Application, called ADDM4RIOTA. More specifically, it is a meta-model to design a resilient IoT Application. The specification of a meta-model with Architectural Design Decisions concepts targeted for resilience providing a common lexicon and taxonomy, defining the main resilient concepts and their relationships to model IoT applications able to handle threats, the restore of operations and adapt to environment changes. Thus generate a common understanding between stakeholders about a target resilient IoT system in question providing an approach that helps to capture precisely state requirements and domain knowledge. ADDM4RIOTA can generate a primary representation of an IoT Application architecture from the point of view of resilience so that group of stakeholders can communicate.

This paper brings four contributions: First, the requirements for modelling a Resilient IoT application are presented. Second, it presents a definition for Resilient IoT Applications based on the resilience requirements raised. Third, a Meta-Model to define a common understanding of a field of interest from the point of view of resilience, through the definition of its vocabulary and key resilient constraints. Fourth, it presents a modelling process to use the ADDM4RIOTA to design of resilient IoT applications; such process allows the separation of responsibilities between the different experts involved in the construction of an IoT application.

This paper is organized as follows. The section 2 presents the requirements for modelling a resilient IoT application. In section 3 the ADDM4RIOTA is described. In section 4, the modelling process for resilient IoT application design is presented. The section 5 illustrate the use and demonstrate the value of the ADDM4RIOTA through modelling of a case. We briefly discuss related work in section 6 and the paper concludes with future work and conclusions in section 7.

2 REQUIREMENTS FOR MODELLING A RESILIENT IOT APPLICATION

Resilient systems can endure and successfully recover from disturbances by identifying problems and mobilizing the available resources to cope with the disturbance. Resiliency techniques allow a system recover from disruptions, variations, and a degradation of expected working conditions [1, 6]. A Biological system such as the Immune System (IS) is resilient [6, 23]. The Immune System is highly adaptive and scalable, with the ability to cope with multiple data sources, fuse information together and making decisions. The IS has multiple interacting agents, operates in a distributed manner over a multiple scale, and has a memory structure to enable learning. The IS is considered a good example of resilient system because it is a complex system that is in operation in most living beings in our planet, and has been improving itself over millions of years through the process of evolution called natural selection [9, 15, 23, 32]. Furthermore, the IS has already inspired some works in computer science [6, 10, 12].

Given the aforementioned advantages of the IS, this paper inspire the resilience requirements for IoT application modelling on the some key resilience properties of the IS. The IS has five resilience key properties: (i) Monitoring, (ii) Detection, (iii) Protection (iv) Restoration and (v) Memorization [32]. In the following will be presented these five key resiliency properties of the Immune System, that will be used as requirements for the resilient IoT application modelling.

Monitoring

As IS property: The Immune System has cells called Leukocytes that are produced or stored in many locations in the body, including the thymus, spleen, and bone marrow. The two basic types of leukocytes are (i) phagocytes, cells that chew up invading organisms and (ii) lymphocytes, cells that allow the body to remember and recognize previous invaders and help the body destroy them. The leukocytes circulate through the body between the organs and nodes via lymphatic vessels and blood vessels. In this way, the immune system works in a coordinated manner to monitor the body for detecting germs or substances that might cause problems [23, 32].

As IoT requirement: Monitoring of the resilient IoT application is an essential requirement. Such monitoring should perform inspection on operational resources, data flow and energy efficiency. The monitoring data should be stored in Knowledge Base for other components such as protection, detection and restoration to retrieve and perform their respective operations. Using an IoT Gateway could be an effective way to monitor the behavior of any system in several layers: application, network and physical [4, 5]. IoT gateways may not be used only for communication, to connect the sensors to the internet, or to collect the data from sensors. Such IoT gateways can perform the function of monitoring the system. For example, if a sensor is damaged, it must be replaced automatically. An IoT gateway is an important aspect of building an efficient, secure, and easy-to-maintain system [13].

Detection

As IS property: The immune system detects the pathogens, and then efficiently eliminating the pathogens by recognizing the molecules of the pathogen and designing other molecules that fit like a lock and key with only the pathogen molecules. These molecules that are made by design carry with them the tools to kill the pathogen and are called antibodies [23, 32].

As IoT requirement: Before dealing with Threats, an IoT application has to identify them. A Threat is any potential danger to IoT systems such as faults, failures and errors. The main IoT Threats can originate from three primary sources: Nature source, Hardware source and Human source. Natural threats, are natural events such as Earthquakes, Hurricanes, Floods, Fire and Power Outage; Hardware threats, are threats that originate from the hardware characteristic, such as, Energy Limitations, Memory Limitations, Natural wear, Malfunctions, Computational Limitations, Mobility, Scalability, Communications Media, Multiplicity of devices and low Battery; and Human threats are those caused by people, such as malicious threats consisting of internal (someone has authorized access) or external threats (individuals or organizations working outside the network) looking to harm and disrupt an IoT Application. Human threats are categorized in function of three layers that IoT Application is typically structured: Application Layer, Network Layer and Physical Layer. Thus, the processes of detection should be initiated by monitoring to find the causes of threats. The data gathered by detection should be stored in one knowledgebase. This data can be utilized by restoration to perform restoration and optimization of application. To perform detection different detection algorithms depending upon the scenario must be used. The threats detected will update the threats list in knowledge base for future protection.

Protection

As IS property: The immune system protects the body against illness and infection caused by external threats, such as bacteria, viruses, fungi or parasites, and internal threats, such as cancer cells. The protection property is performed by redundant cells and a collection of reactions that the body makes in response to cancer cells or infection [23, 32].

As IoT requirement: Protection should play both the role of defense and offence in an IoT application against main threats that can occur. The defense part is to protect from threat and should keep updating the knowledge base for new threats. The offence part is that if application has been harmed by some threat in the past and the same attack occurs again; it should protect the application from it. To perform protection could be used self-protection and redundancy that also could be used to perform restoration. Self-protection relates to the protection of the overall system from threats. Overall failure can be either an effect of a malicious attack or of a cascade of failures experienced by a whole series of system components. Self-protection not only includes countermeasures against single failures or attacks to influence the behaviour of the whole system, but also to anticipate such situations and to try to avoid them [7]. There is not much that has to be said to motivate the importance of self-protection in the context of IoT. Redundancy is one of the most fundamental approaches to achieving resilience. In most cases, redundancy refers to deploying into a system more components than are required for the functionality. Functional redundancy refers to functional overlap of different components. Functional redundancy makes it possible to substitute a failed component with a different one in order to recover the whole or part of lost functionality [27].

Restoration

As IS property: While the IS is identifying pathogens, and eliminating them, it makes the body work with minimal resources until it can recover the body to stable state. This is done through complex and ordered processes such as inflammation and healing [23, 32].

As IoT requirement: Restoration should bring back the IoT application to its normal state after catastrophic situation. The restoration component should perform healing on weakened parts of the application and empower such parts to perform their regular functions. The threats should be detected by the detection component with the help of the monitoring component. To perform restoration could be used Redundancy, Self-Configuration, Self-Healing, Fault-recovery, and Disaster recovery strategies. Dynamic reconfiguration of the component should be considered, when the workload is increased, to achieve better application performance. Self-Optimization also should be used in case when full restoration is not achieved by the restoration. Some architectural constraints will now be described: i) Self-Configuration allows the system be capable to readjust itself. Readjustment of the system is required if its environment changes or to reach an objective set for the system [7]. ii) Self-optimization allows the system can measure its current performance and it able to compare it against to the known optimum level of performance. The system will adjust its operation to reach closer the optimal performance. The system is also able to change its operation to cope with new user set policies [7]. iii) Self-healing allows the system tries to recover from faults or to avoid them. Self-healing can be implemented in two different styles. They are reactive and proactive modes. In reactive mode the system detects and recovers from faults as they occur. The system also tries to repair the faulted functions if possible. In proactive mode the system monitors its state to detect and adjust its behaviour before reaching an undesired state [7].

Memorization

As IS property: It is the ability of the immune system to respond more rapidly and effectively to pathogens that have been encountered previously, and reflects the pre-existence of a clonally expanded population of antigen specific lymphocytes [23, 32].

As IoT requirement: The knowledge base component is memory of resilient IoT Application. It is a database that keeps the information such as monitoring data, restoration data, vulnerabilities list, and information about all types of attack and its precautionary measures. Monitoring data is collected by the gateway working with all the components. This data will be used by other components to perform their operations. Detection data is gathered during detection phase. This data includes the detected faults and this data could be used by restoration to perform restoration of resilient IoT application.

3 ADDM4RIOTA

Before describing the ADDM4RIOTA we will present our definition for Resilient IoT Application that was used to develop it. Because despite the considerable interest from the industry and academia, there is no consensus about one definition. Thus, in view of the resilience requirements raised in the previous section, and also on the definition of IoT as introduced by [16] and on the definition of the Resiliency of system as introduced by [15], we adopt the definition of Resilient IoT application as;

Group of infrastructures interconnecting connected objects and allowing their management, data mining and the access to the data they generate to reach common goals with capability 1) monitoring the system constantly, 2) to detect new and old threats that can damage the system, 3) to protect the application resisting external and internal threats, 4) to recover to stable state and/or adapt structure to work with minimum resources and 5) to memorize all the impacts that threats can cause on IoT Services, Resources and Devices to allow faster and more effective responses to deal with future threats.

The ADDM4RIOTA was developed aiming to attend this definition and help reduce the difficulty to design of Resilient IoT application. For this, the meta-model proposed brings: i) the main threats described in the literature to speed up the threats identification procedure in the design process; ii) the tactics, constraints and properties of resilience represented as first class to make explicit and allow to capture potential alternative resilient solutions, iii) architectural design decisions principles, such as Issues (IoT Threats), Solutions (Resilient Countermeasures) and Decisions, to support the decision of select or reject resilient countermeasures to mitigate the threats; and iv) Group decision Making principles to driving the way stakeholders make collaborative decisions. The ADDM4RIOTA is divided into four packages. This was done in order to facilitate the understanding and use of the meta-model. The four packages are: Inputs (colorful elements in dark orange), Issues (colorful elements in red), Countermeasures (colorful elements in green) and Decisions (colorful elements in yellow). The Figure 1 depicts all packages, the main elements and the relationships between them. The Inputs package contains classes to represent the elements of an IoT application domain model such as IoT Critical Objects that are affected by IoT Threats. The Issues package formalizes the concept IoT Threats and request Resilient Countermeasures as possible solution. The Countermeasures package describes the concept of a Resilient Countermeasures that mitigates IoT Threats. Lastly, the Decisions package formalizes the decisions concept to select and reject a resilient countermeasure for addresses IoT Threats. The following will explain in detail each package.

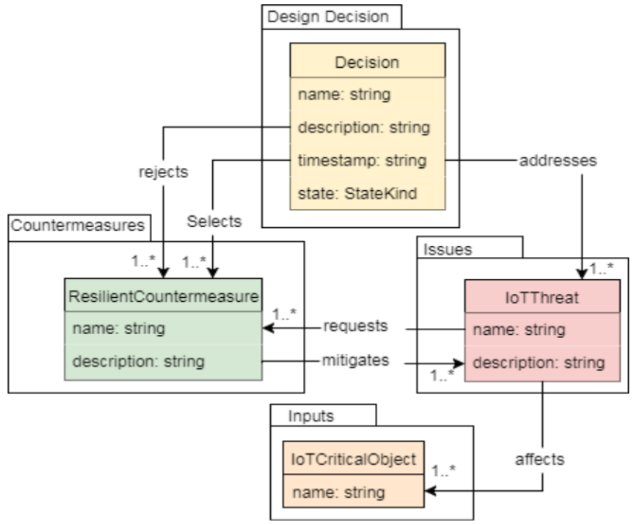

Figure 1: The four packages that composed the ADDM4RIOTA.

Input Package

The Inputs package is described in Figure 2. This package defines all ADDM4RIOTA inputs as IoT critical objects. It represents an element that can be affected by IoT threat and was proposed in [24]. These IoT critical objects can be identified in an IoT application domain model. It can be generated by any domain meta-model, such as IoT Domain Model (IoT-DM) and Personalized Monitoring System Domain Model (PMS-DM) [7, 20]. Through the Input package we can identify the critical objects in an IoT application domain model identifying the elements in an application that have the highest chance of being affected by a threat and therefore damages the functioning of the system. A Critical Object can be IoT Hardware and IoT Software components [7]. The IoT Hardware components are classified as: Device, Tag, Sensor and Actuator. The IoT software components are classified as: Active Digital Artefacts, Passive Digital Artefact, Service, Resource, Network Resource and On-Device Resource. Details about these components can be found in [7, 20]

Issue Package

The Issues package is highlighted in Figure 3 and has 15 elements. This package brings a set of concepts, entities and relationships that allow describing the main IoT threats that can damage an IoT application. In ADDM4RIOTA a IoT Threat is an action that takes advantage of security weaknesses in an IoT Application and has a negative impact on it. The Motivation and Cause elements describe the IoT Threat. A Motivation why the IoT Threat is a problem, and the Cause of this IoT Threat. IoT Threats can originate from three primary sources: Nature source, Hardware source and Human source. In [38] can be found a detailed description of each of these threats type that compose the enumerations: Hardware Threat Type, Nature Threat Type, Application Layer Threat Type, Network Layer Threat Type and Physical Layer Threat Type.

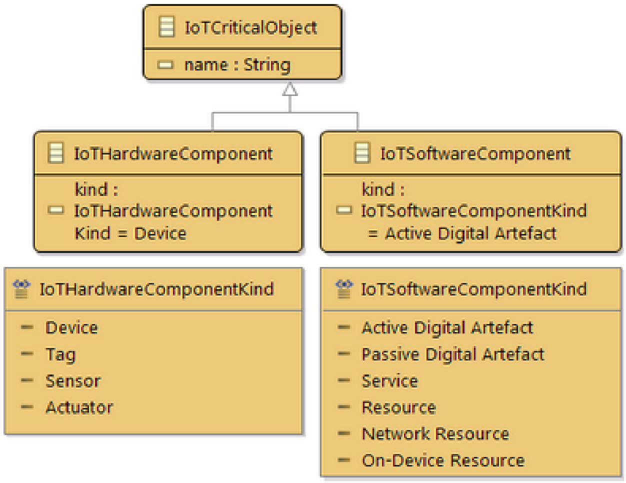

Figure 2: Input Package.

Countermeasures Package

This Package is highlighted in Figure 4 and has 24 elements. It brings together a set of technologies available to implement the resilience properties that enable an IoT application to handle an IoT Threat. This Package has five fundamental properties that allow addressing the definition of Resilient IoT Application that are: Monitoring, Protections, Detection, Restoration and Memorization (knowledge base).

A Resilient Countermeasure can be classified in four properties: Monitoring, Protections, Detection and Restoration and interacts with knowledge base.

Monitoring: it performs monitoring on Operational resources, Data flow, Energy Efficiency components and help to protection, detection and restoration to work. The Monitoring of IoT application can be performed by a Gateway. The monitoring data will be stored in Knowledge base for other resilient solution to retrieve and perform the necessitated operations. The monitoring of IoT Application through IoT gateways can be performed using Autonomic Architectures (the architectures that compose the enumeration called Autonomic Architectures kind enumeration are described in [38]).

Protection: it can be implemented through 9 tactics of Redundancy and 29 Self-protection techniques. The Protection update knowledge base when, for example, get a new attacks

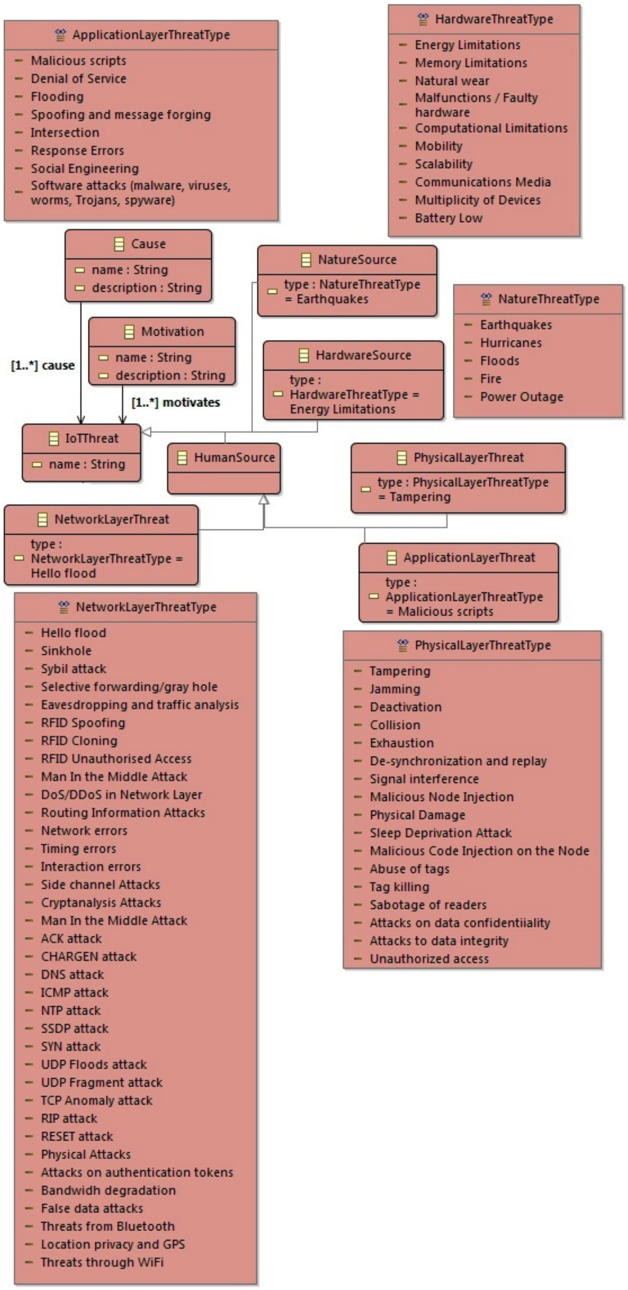

Figure 3: Issues Package.

and retrieve old situation from knowledge base. The tactics that compose the enumeration Redundancy Technique Kind and Self Protection Technique Kind are described in [38].

Detection: it detects the vulnerabilities and the weak points of the Resilient IoT Application. The processes of detection is performed by monitoring through the implementation of Detection Techniques (all techniques that compose the enumeration Detection Technique Kind are described in [38]) to find some IoT Threat. Detection techniques are algorithms and the vulnerabilities detected will be stored in the vulnerability list in knowledge base for future Protection and can be utilized by Restoration resilient solution in order to perform restoration and optimization of Resilient IoT Application.

Restoration: the main responsibility of it is to bring back the Resilient IoT Application to its normal state after catastrophic situation. The restoration will perform Self-Healing on weakened parts of the Resilient IoT Application and empower them to perform their regular functions. The weak parts will be detected by the detection resilient solution with the help of the monitoring resilient solution. Self-Configuration will be used to reconfigure the components when workload is increased to achieve optimization of Resilient IoT Application. Self-Optimization will also be used in case when full restoration is not achieved by the restoration. The Restoration resilient solution can implement Disaster Recovery Strategy like backup and contingency plans that are the best approaches to secure systems against natural threats. Finally, the restoration resilient solution can implement too Fault Recovery techniques important to deal with IoT Threats in WSN. All techniques to implement Restoration that compose the enumeration: Self Configuration Technique Kind, Self-Healing Technique Kind, Self-Optimization Technique Kind, Fault Recovery Technique Kind, Disaster Recovery Strategy Technique Kind are described in [38].

Decision Package

The Decision Package is highlighted in Figure 1. This Package combines Architecture design decisions and Group decision making principles and methods to enable group of stakeholders to find the best resilient countermeasure to be to address an IoT Threat identified by the Issues Package. This Package allows a model instantiated from ADDM4RIOTA be a primary representation of the architecture. A primary representation of architecture consists of architectural decisions and good architecture results from making good architectural decisions [35]. Decision package has 13 elements and the main classes of this package is Decision, from is possible select or reject a resilient countermeasure to addresses an IoT Threat. The other elements of Decision Package use the same concepts of the two meta-models in the literature the Archium meta-model [22] and Architecture design decisions Meta-model with Group Decision making [28]. Due to limited space they are explain in details in [38].

4 MODELLING PROCESS

This section presents the steps required to design a resilient IoT application using the ADDM4RIOTA. It is divided into four phases that are executed in an iterative way as depicted

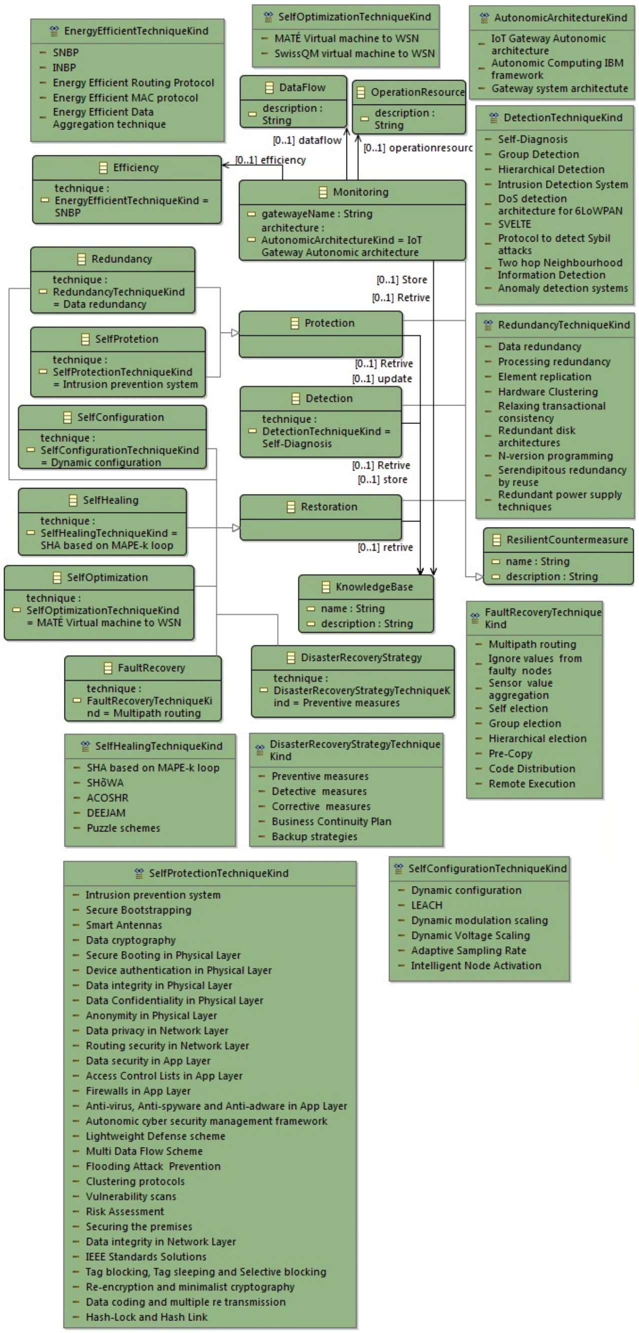
EnergyEfficientTechniqueKind
SNBP
INBP
Energy Efficient Routing Protocol
Energy Efficient MAC protocol
Energy Efficient Data Aggregation technique
SelfOptimizationTechniqueKind
MATÉ Virtual machine to WSN
SwissQM virtual machine to WSN
AutonomicArchitectureKind
IoT Gateway Autonomic architecture
Autonomic Computing IBM framework
Gateway system architectute
DataFlow
description : String
OperationResource
description : String
[0..1] dataflow
[0..1] operationresourc
DetectionTechniqueKind
Self-Diagnosis
Group Detection
Hierarchical Detection
Intrusion Detection System
DoS detection architecture for 6LoWPAN
SVELTE
Protocol to detect Sybil attacks
Two hop Neighbourhood Information Detection
Anomaly detection systems
Efficiency
technique : EnergyEfficientTechniqueKind = SNBP
[0..1] efficiency
Monitoring
gatewayeName : String
architecture : AutonomicArchitectureKind = IoT Gateway Autonomic architecture
Redundancy
technique : RedundancyTechniqueKind = Data redundancy
[0..1] Store
[0..1] Retrive
Protection
SelfProtetion
technique : SelfProtectionTechniqueKind = Intrusion prevention system
[0..1] Retrive
[0..1] update
RedundancyTechniqueKind
Data redundancy
Processing redundancy
Element replication
Hardware Clustering
Relaxing transactional consistency
Redundant disk architectures
N-version programming
Serendipitous redundancy by reuse
Redundant power supply techniques
Detection
technique : DetectionTechniqueKind = Self-Diagnosis
SelfConfiguration
technique : SelfConfigurationTechniqueKind = Dynamic configuration
[0..1] Retrive
[0..1] store
SelfHealing
technique : SelfHealingTechniqueKind = SHA based on MAPE-k loop
Restoration
[0..1] retrive
ResilientCountermeasure
name : String
description : String
SelfOptimization
technique : SelfOptimizationTechniqueKind = MATÉ Virtual machine to WSN
KnowledgeBase
name : String
description : String
FaultRecoveryTechniqueKind
Multipath routing
Ignore values from faulty nodes
Sensor value aggregation
Self election
Group election
Hierarchical election
Pre-Copy
Code Distribution
Remote Execution
FaultRecovery
technique : FaultRecoveryTechniqueKind = Multipath routing
DisasterRecoveryStrategy
technique : DisasterRecoveryStrategyTechniqueKind = Preventive measures
SelfHealingTechniqueKind
SHA based on MAPE-k loop
SHõWA
ACOSHR
DEEJAM
Puzzle schemes
DisasterRecoveryStrategyTechniqueKind
Preventive measures
Detective measures
Corrective measures
Business Continuity Plan
Backup strategies
SelfProtectionTechniqueKind
Intrusion prevention system
Secure Bootstrapping
Smart Antennas
Data cryptography
Secure Booting in Physical Layer
Device authentication in Physical Layer
Data integrity in Physical Layer
Data Confidentiality in Physical Layer
Anonymity in Physical Layer
Data privacy in Network Layer
Routing security in Network Layer
Data security in App Layer
Access Control Lists in App Layer
Firewalls in App Layer
Anti-virus, Anti-spyware and Anti-adware in App Layer
Autonomic cyber security management framework
Lightweight Defense scheme
Multi Data Flow Scheme
Flooding Attack Prevention
Clustering protocols
Vulnerability scans
Risk Assessment
Securing the premises
Data integrity in Network Layer
IEEE Standards Solutions
Tag blocking, Tag sleeping and Selective blocking
Re-encryption and minimalist cryptography
Data coding and multiple re transmission
Hash-Lock and Hash Link
SelfConfigurationTechniqueKind
Dynamic configuration
LEACH
Dynamic modulation scaling
Dynamic Voltage Scaling
Adaptive Sampling Rate
Intelligent Node Activation

Figure 4: Countermeasures Package.

in Figure 5. The activities are performed by the main group of IoT application stakeholders. The set of actors in our process is composed of:

Domain expert: responsible for instantiates the IoT Domain model by identifying the domain elements, such as virtual entities, resources, devices, services and users. It has ability to understand domain concepts, including the data types produced by the sensors, consumed by actuators, accessed from storages, user interactions, and how the system is divided into regions.

Resilience Expert: responsible for identifying the critical objects in IoT Domain Model and lists the threats and associated countermeasures. It has experience on fault, failures and error in IoT device and software, and knowledge on security, self-management and resilient constraints.

Device developer: responsible for writing drivers for the sensors, actuators, storages, and end-user applications used in the domain. It has a deep understanding of the inputs/outputs, and protocols of the individual devices.

Software designer: responsible for defining the structure of an IoT application by specifying the software components and their generate, consume, and command relationships. It has Software architecture concepts, including the proper use of interaction modes such as publish/subscribe, command, and request/response for use in the application.

Network Manager: responsible for install the application on the system at hand; this process may involve the generation of binaries or bytecode, and configuring middleware. It has a deep understanding of the specific target area where the application is to be deployed.

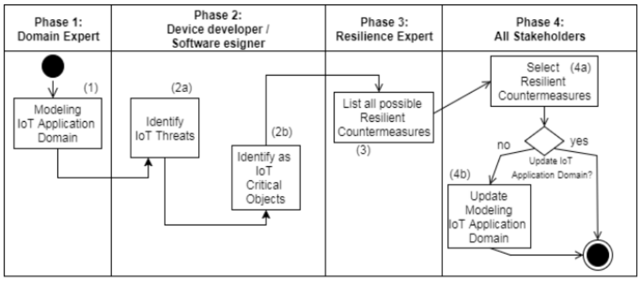
Phase 1: Domain Expert
Phase 2: Device developer / Software esigner
Phase 3: Resilience Expert
Phase 4: All Stakeholders
(1)
Modeling IoT Application Domain
(2a)
Identify IoT Threats
(2b)
Identify as IoT Critical Objects
List all possible Resilient Countermeasures
(3)
Select Resilient Countermeasures
(4a)
no
yes
Update IoT Application Domain?
(4b)
Update Modeling IoT Application Domain

Figure 5: UML Activity Diagram describing the ADDM4RIOTA modelling process.

Figure 5 depicts the UML Activity Diagram illustrating the modelling process with its phases and actors (or stakeholders).

The first phase of the modelling process (Phase 1) encompasses the modelling IoT Application (activity 1a). This activity is performed by Domain Expert that should instantiate the application domain model, since from it will be identified the IoT Threats and IoT Critical Objects. For this activity could be used any Domain meta-model of literature such as IoT Domain Model (IoT-DM) [7] and Personalized Monitoring System Domain Model (PMS-DM) [20].

The second phase of the modelling process (Phase 2) encompasses two activities: Identify IoT Threats (activity 2a) and IoT Critical Objects (activity 2b). These activities will be performed by Device developer and Software designer and to help them carry out these activities we provide a table called *Relationship between IoT Application Domains, IoT Critical Objects and IoT Threats* (available in [38]) based on [18, 21, 26]. It gathers the main IoT Threats and critical IoT Objects for the three main domain of IoT Application: Industrial, Smart city and Health well-being. In this table we have the relation between the main elements that can be classified as IoT Critical Objects and the main source of IoT Threats that can affect the workings of these elements. Furthermore, the *Relationship between IoT Application Domains, IoT Critical Objects and IoT Threats* table has relation with *IoT Threats* Tables available in [38].

The third phase of the modelling process (phase 3) encompasses one activity: List possible Resilient Countermeasures (activity 3). It will be performed by Resilience Expert that should list the main resilient countermeasures to mitigate the IoT threats identified in the previous phase. To help the Resilience Expert carry out this activity the enumeration tables of ADDM4RIOTA (available in [38]) gathers the main Resilient Countermeasures. Furthermore, the *IoT Threats Tables* has relation with *Resilient Countermeasures tables*. This relation is of many to many.

The fourth phase encompasses two activities. Select the Resilient Countermeasures (activity 4a). In this activity all stakeholders will participate and use architecture design decisions and group decision using principles and methods through Decision package. It will help to find the best resilient Countermeasures from of the selected in the previous phase, in activity (3). Others stakeholders can be included in the group to participate in modelling processes such as software architects, developers, designers, testers, users, etc. Update modelling IoT Application Domain Model (activity 4b). Before finished the modelling processes must be checked if there is need update in IoT Application due to selection made in activity 4a.

5 CASE: RESILIENT NURSING HOME IOT APPLICATION

In this section, we validate our approach by applying it on a case. It is used to illustrate the utilization of ADDM4RIOTA modelling process to generate a primary representation of a Resilient Nursing Home IoT Application architecture. First, the case is introduced, after the modelling processes of resilient IoT application case is presented in more detail.

Case Overview

The nursing home IoT application case was inspired in [2]. Our case presents the design of an IoT application that aims to perform early detection, rapid, and appropriate response to help in the monitoring of patients who are under care in separate rooms into a nursing home. Therefore, this application also must be resilient because any failure of the system can cause serious damage to patients. The application includes sensors that are being used by patients to capture vital signs and alerts when signals outside the normal patterns are detected. For example, if occurs an abnormal increase in body temperature (that can indicate some infection), the medical staff receive one alert.

Modelling of Resilient Nursing Home IoT Application

Phase 1: This phase encompasses the activity related to modelling of Nursing home IoT application domain. **Activity 1:** This modelling was done using the IoT Domain Meta-model of IoT-ARM [7] and is depicted in Figure 6. The element HumanUser represents the medical team that subscribes to the alarm service, provided by the system through an Android app or a desktop application. The Android and Desktop applications are Active Digital Artefacts (from the IoT Domain Meta-Model) and invoke their respective alarm services: Alarm Panel and Alarm message to alert the medical staff in case of abnormal situation with patients. Alarms represent the communication system interface with the user. The Alarm Panel service is invoked by the desktop computer application and displays an alarm on the PC screen. On the other hand, the Alarm message service is invoked by the Android application and displays a message on the mobile screen of the users that make up the medical team. The Android and Desktop applications to invoke the alarm services them subscribes another service called Human Vital Data Measurement that will read the user vital data from the Database and evaluate if they are out of normal ranges for human health. The database is represented by an element of type Network Resource and is associated with an element of type PassiveDigitalArtefact called Vital Sign Data that represent a Physical Entity called Patients. The vital data captured by sensors are inserted in the database through a service called Store Vital Data that receives the data from the resource Human Vital data, which is an element of type On-DeviceResource that is hosted within the Device. In this case, the On-DeviceResource is a software component that provides a way to connect to the data obtained by the sensors. For example, this data can be exposed through an XBee/ZigBee network. The Device is represented by a microcontroller board, for example, an Arduino, which is connected to the three types of sensors that are collecting the vital patient data. The three types of sensors are: Blood Pressure Sensor, Heart Rate Sensor, and Body Temperature Sensor.

Phase 2: This phase raises the issues that can damage the Nursing Home IoT application. **Activity 2a:** The Software attacks and Malfunction/Faulty hardware are the threats that will be

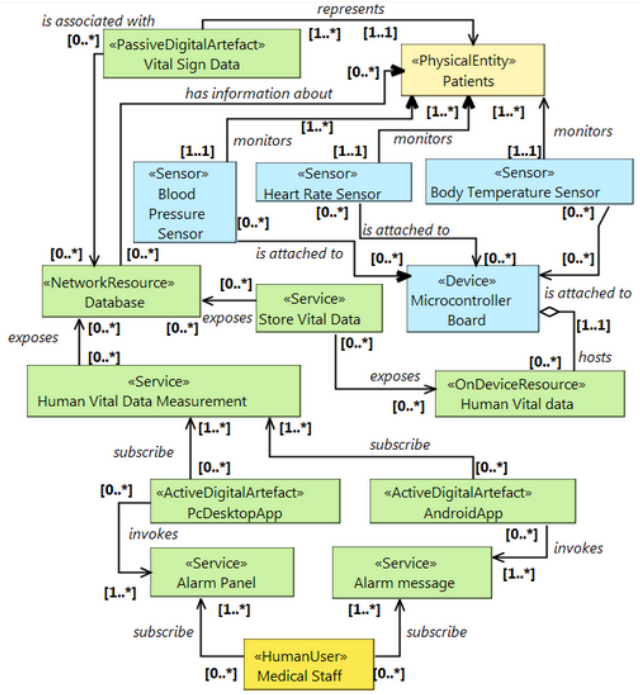

Figure 6: Nursing Home IoT Application Domain Model instantiated from IoT Domain Meta-Model [7].

addressed in Nursing Home IoT application. i) The Software attacks can occur due to Negligence of medical staff, because changes and updates in desktop computer configuration can cause system malfunctioning. Medical staff may install contaminated software upgrades that propagates virus into the desktop computer. A cause of this can be lack of IT team to take care of security and the work overload of medical staff. ii) The Malfunctions/Faulty hardware can occur due to incorrect use, because improper use for a long period can lead to a malfunction and can cause interruption of availability of application. A motivation of this is hiring a new member for medical team. A cause of this can be Lack of training. This IoT Threats are represented by the colored elements in red in the Figure 7. **Activity 2b:** A total of six elements were classified as IoT Critical Objects in function of IoT threats selected in previous activity and with the help of table called *Relationship between IoT Application Domains, IoT Critical Objects and IoT Threats* available in [38]. The elements that were identified as IoT critical object are: i) Active Digital Artefacts called PcDesktopApp and AndroidApp and ii) Device called Microcontroller Board and the Sensors called Blood Pressure, Heart Rate and Body Temperature. These are the main objects that can be affected by the identified threats in the activity 2a. See colorful elements in orange on Figure 7.

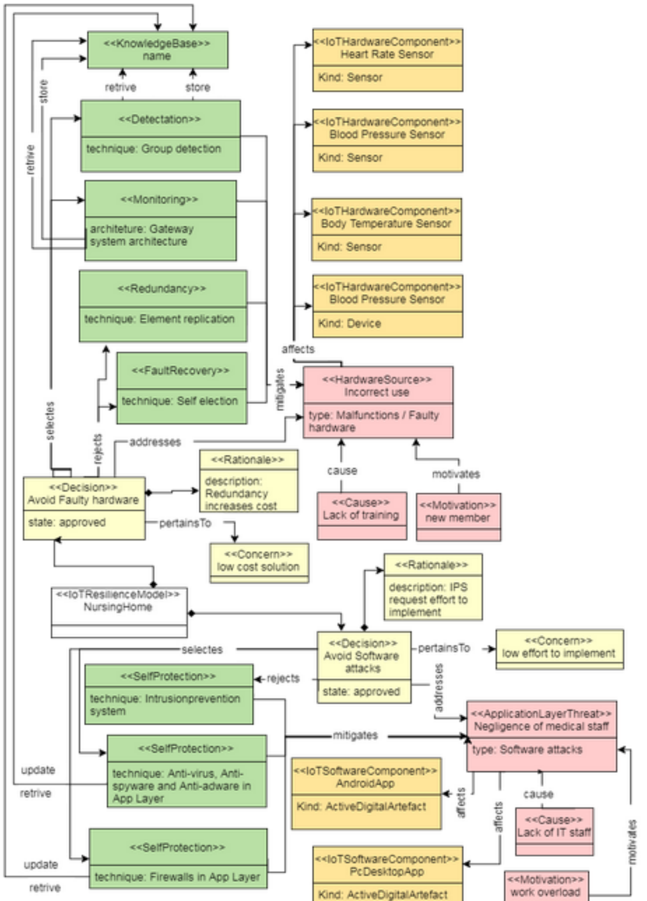

Figure 7: Resilient Nursing Home IoT Application Model instantiated from ADDM4RIOTA.

Phase 3: this phase raises the some possibles Resilient Countermeasures in functions of Threats and critical object identified in the Nursing Home IoT application domain model. **Activity3:** Selection made based on the enumeration tables available in [38], where the references and the explanation for each tactic can be found. The Figure 7 exposes the countermeasures (see colorful elements in green).

Four countermeasures were selected to mitigate the Malfunctions/Faulty hardware that the Incorrect use of sensors and devices can cause in application.

i) Monitoring using IoT Gateway Autonomic architecture. In [19] was presented an intelligent architecture which consists of a large number of sensing objects for monitoring purposes that can be used in IoT application. An embedded-based gateway for use in a monitoring system was proposed in an IoT network. The gateway is a critical component for collecting, recording and forwarding data obtained from sensors. It is programmable, low-cost, real-time and flexible. The software and hardware for wired and wireless communication interfaces are successful and suitable for field trials.

ii) Detection using Group Detection. The goal of fault detection is to verify that the services being provided are functioning properly, and in some cases to predict if they will continue to function properly in the near future. In [24] a detection mechanism is proposed to identify faulty sensor nodes. Algorithm is based on the idea that sensors from the same region should have similar values unless a node is at the boundary of the event-region. The algorithm start by taking measurements of all neighbors of a node and uses the results to calculate the probability of the node being faulty

iii) Redundancy using Element replication. The structure and functionality of a replica are exactly the same as that element so that they can substitute each other without problem [7]. In case of Nursing Home IoT application the sensors and device could be replicated.

iv) Fault Recovery using Self-Election. When passive replication is applied, the primary replica receives all requests and processes them. In order to maintain reliability between replicas, the state of the primary replica and the request information are transferred to the backup replicas atraves de self-election [31].

Three countermeasures were selected for mitigate Software attacks that Negligence of medical staff can allow that affect the software component of application.

i) Self-Protection using Intrusion prevention system. To follow this tactic the Cumulative-Sum-based Intrusion Prevention System (CSIPS) can be applied. It which detects malicious behaviors, attacks and distributed attacks launched to remote clients and local hosts based on the Cumulative Sum (CUSUM) algorithm [25].

ii) Self-Protection using Anti-virus, Anti-spyware and anti-adware in application layer. Security software like antivirus or anti spyware is important for the reliability, security, integrity and confidentiality of the IoT system and desktop computer.

ii) Self-Protection using Firewalls in application layer. This is an extra effective layer of security that will help block attacks that authentication, encryption and ACLs would fail to do so. Authentication and encryption passwords can be broken if weak passwords were selected. A firewall can filter packets as they are received, blocking unwanted packets, unfriendly login attempts, and DoS attacks before even authentication process begins.

All these countermeasures are related to the knowledge base. It is a resilience requirement that the application has memorization of all occurrences.

Phase 4: Select the countermeasures that best fit the concerns of the Resilient Nursing Home IoT application stakeholders.

Activity 4a: Here the stakeholders of the case made two decisions: i) the Decision to Avoid Malfunctions/Faulty hardware was to select Detection using Group detection technique and Monitoring using Gateway system architecture as resilient countermeasures. Because to achieve the concern of low cost solution the rationale is reject Redundancy using Element replication and Fault Recovery using self-election. Since this increase in the number of sensors and devices and thus increases the cost of the project. ii) the Decision to Avoid software attacks was to select self-protection using Anti-virus, Anti-spyware and Anti-adware in application layer and self-protection using Firewalls in application layer as resilient countermeasures. Because to achieve the concern of low effort to implement this solutions in the application the rationale is reject Self-protection using Intrusion Prevention system. Since the implementation will require the use of complex algorithms that will order more qualified workers in project. **Activity 4b:** Update in Nursing Home IoT application Domain Model. As a gateway is going to be used it will be necessary to update the model by inserting this new element.

6 RELATED WORK

Some projects try to address the challenge of deal of resilience in Iot Application. But them present some drawbacks. Here we will highlight three important projects developing IoT architectures [17, 36].

IoT-A [7] is a European project that proposes an Architectural Reference Model (ARM) for IoT. But the IoT ARM is not an IoT architecture per se, but a set of best practices, guide-lines, and a starting point to generate specific IoT architectures [17]. The unique resiliency treatment presented by IoT-ARM is through an architectural perspective called Availability and Resilience in which it presents a design choice catalog with only 9 generic tactics used in software architecture. The ADDM4RIOTA presents a total of 82 tactics to implement the resilient constraints such as redundancy, self-configure, self-heal, self-optimise and self-protect specific for IoT application architecture. BeTaaS [30] is a project that proposes an architecture for the IoT and Machine-to-machine (M2M) communication, to enable running applications over a local cloud of gateways. Betaas focuses in Dependability, but presents aspect of resilience. It is handled via the Failure Analysis Approach component that is responsible for the identification of potential causes of failures and for providing solutions to properly manage them. However presents no concept of ADDs and GDM in order to select possible solutions for failures as well as ADDM4RIOTA features. In addition to not specifying which elements can and should be classified as critical. The EU FP7 OpenIoT research project, has introduced an IoT architecture [11]. OpenIoT is based on IoT-ARM to achieve alignment, architecture development and specification. OpenIoT Address resilience only partly and places the focus on resilience in terms of mitigation. For that, OpenIoT maintains an up-to-date inventory of entities and dynamically restructures the dependencies between entities, e.g., reconnects a service to another sensor in case of sensor failure. Thus, fail-over and recovery are integral parts of OpenIoT.

The projects above mentioned are solutions to specific problems of system information and do not consider the specification of resilience for IoT application. Furthermore, they do not provide elements to express resilience accurately in the early stages of development. They do not provide modelling mechanisms that contemplate resilience as first class representation to design a resilient IoT application like ADDM4RIOTA.

7 CONCLUSION

In this paper, we presented an Architectural Design Decision Model for Resilient IoT Application, called ADDM4RIOTA, due to high susceptible to threats of IoT Application and lack of modelling approaches contemplating resilience as first class representation to design Resilient IoT Application. It provides a common lexicon and taxonomy, defining the main resilient concepts and their relationships, and a modelling process needed to generate a common understanding and facilitate decision making between stakeholders about a target resilient IoT application in question. The ADDM4RIOTA concepts were exemplified with the modelling of a Resilient Nursing Home IoT Application. The modelling of the case allow to note how ADDM4RIOTA can reduces the difficulty of design an IoT application with resilient concepts. The ADDM4RIOTA generated a primary representation of Resilient Nursing Home IoT Application architecture so that group of stakeholders were able to communicate. The ADDM4RIOTA allowed to identify IoT Critical Objects in Nursing Home IoT Application domain and IoT Threats that could affect them. Next, it was possible to find possible Resilient Countermeasures in functions of IoT Threats and IoT Critical Objects, and select which would be the best ones for the case. In the future work, we intend to integrate ADDM4RIOTA in a framework for a Design Process Flow approach with a domain model, information model, algorithms and ERD diagrams to assist with more artifacts the design process of IoT Application or Personalized Monitoring System.

REFERENCES

[1] Adnan Ahmed and Syed Shahram Hussain. 2007. Meta-model of resilient information system.

[2] Aintzane Armentia, Unai Gangoiti, Rafael Priego, Elisabet Estévez, and Marga Marcos. 2015. Flexibility support for homecare applications based on models and multi-agent technology. *Sensors* 15, 12 (2015), 31939–31964.

[3] Qazi Mamoon Ashraf and Mohamed Hadi Habaebi. 2015. Autonomic schemes for threat mitigation in Internet of Things. *Journal of Network and Computer Applications* 49 (2015), 112–127.

[4] QAZI MAMOON Ashraf and MOHAMED HADI Habaebi. 2015. Introducing autonomy in internet of things. *Recent Advances in Computer Science, WSEAS Publishing* (2015), 215–221.

[5] Qazi Mamoon Ashraf, Mohamed Hadi Habaebi, Gopinath Rao Sinniah, Musse Mohamud Ahmed, Sheroz Khan, and Shihab Hameed. 2014. Autonomic protocol and architecture for devices in Internet of Things. In *2014 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA)*. IEEE, 737–742.

[6] Marco Autili, Amleto Di Salle, Francesco Gallo, Alexander Perucci, and Massimo Tivoli. 2015. Biological Immunity and Software Resilience: Two Faces of the Same Coin?. In *International Workshop on Software Engineering for Resilient Systems*. Springer, 1–15.

[7] Alessandro Bassi, Martin Bauer, Martin Fiedler, and Rob van Kranenburg. 2013. *Enabling things to talk*. Springer-Verlag GmbH.

[8] Eleonora Borgia. 2014. The Internet of Things vision: Key features, applications and open issues. *Computer Communications* 54 (2014), 1–31.

[9] Ashik Chandra. 2010. Synergy between biology and systems resilience. (2010).

[10] Petar Čisar, Sanja Maravić Čisar, and Branko Markoski. 2014. Implementation of immunological algorithms in solving optimization problems. *Acta Polytechnica Hungarica* 11, 4 (2014).

[11] OpenIoT Consortium et al. 2013. OPENIoT project description.

[12] Dipankar Dasgupta and Nii Attoh-Okine. 1997. Immunity-based systems: A survey. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, Vol. 1. IEEE, 369–374.

[13] Soumya Kanti Datta, Christian Bonnet, and Navid Nikaein. 2014. An IoT gateway centric architecture to provide novel M2M services. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE, 514–519.

[14] Sofie De Rouck, An Jacobs, and Mark Leys. 2008. A methodology for shifting the focus of e-health support design onto user needs: a case in the homecare field. *International journal of medical informatics* 77, 9 (2008), 589–601.

[15] Kemal A Delic. 2016. On resilience of iot systems: The internet of things (ubiquity symposium). *Ubiquity* 2016, February (2016), 1.

[16] Bruno Dorsemaine, Jean-Philippe Gaulier, jean-philippe Wary, Nizar Kheir, and Pascal Urien. 2015. Internet of Things: A Definition Taxonomy. https://doi.org/10.1109/NGMAST.2015.71

[17] Vangelis Gazis, Manuel Goertz, Marco Huber, Alessandro Leonardi, Kostas Mathioudakis, Alexander Wiesmaier, and Florian Zeiger. 2015. Short paper: IoT: Challenges, projects, architectures. In *2015 18th International Conference on Intelligence in Next Generation Networks*. IEEE, 145–147.

[18] Kashif Habib and Wolfgang Leister. 2015. Threats identification for the smart internet of things in eHealth and adaptive security countermeasures. In *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 1–5.

[19] Ji-De Huang and Han-Chuan Hsieh. 2013. Design of gateway for monitoring system in IoT networks. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. IEEE, 1876–1880.

[20] Cristovao Iglesias, Claudio Miceli, and David Silva. 2019. A Domain Model for Personalized Monitoring System Based on Context-Aware Data Fusion. In *2019 22nd International Conference on Information Fusion (FUSION)*. IEEE, 1–5.

[21] Sidra Ijaz, Munam Ali Shah, Abid Khan, and Mansoor Ahmed. 2016. Smart cities: A survey on security concerns. *International Journal of Advanced Computer Science and Applications* 7, 2 (2016), 612–625.

[22] Anton Jansen and Jan Bosch. 2005. Software architecture as a set of architectural design decisions. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*. IEEE, 109–120.

[23] Chris A Kaiser, Monty Krieger, Harvey Lodish, and Arnold Berk. 2007. *Molecular cell biology.* WH Freeman.

[24] Bhaskar Krishnamachari and Sitharama Iyengar. 2004. Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Trans. Comput.* 3 (2004), 241–250.

[25] Fang-Yie Leu and Zhi-Yang Li. 2009. Detecting DoS and DDoS attacks by using an intrusion detection and remote prevention system. In *2009 Fifth International Conference on Information Assurance and Security*, Vol. 2. IEEE, 251–254.

[26] Huichen Lin and Neil Bergmann. 2016. IoT privacy and security challenges for smart home environments. *Information* 7, 3 (2016), 44.

[27] Dong Liu, Ralph Deters, and Wen-Jun Zhang. 2010. Architectural design for resilience. *Enterprise Information Systems* 4, 2 (2010), 137–152.

[28] Ivano Malavolta, Henry Muccini, and Smrithi Rekha. 2014. Enhancing architecture design decisions evolution with group decision making principles. In *International Workshop on Software Engineering for Resilient Systems*. Springer, 9–23.

[29] Friedemann Mattern and Christian Floerkemeier. 2010. From the Internet of Computers to the Internet of Things. In *From active data management to event-based systems and more*. Springer, 242–259.

[30] Enzo Mingozzi. 2013. BETaaS: Building the Environment for the Things as a Service. In *4th ETSI M2M Workshop*.

[31] Pramod Nagalgaonkar, Dhanraj Biradar, and Gaikwad Ranjit Sharnappa. 2015. Review on Fault Detection and Recovery in WSN. *International Journal of Advanced Research inComputer Science and Software Engineering* 5 (08 2015).

[32] Peter Parham. 2014. *The immune system.* Garland Science.

[33] Hunor Sándor, Béla Genge, and Gheorghe Sebestyén-Pál. 2015. Resilience in the Internet of Things: The software defined networking approach. In *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 545–552.

[34] Mojtaba Shahin, Peng Liang, and Mohammad Reza Khayyambashi. 2009. Architectural design decision: Existing models and tools. In *2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture*. IEEE, 293–296.

[35] Jeff Tyree. 2005. Architectural design decisions session report. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*. IEEE, 285–286.

[36] Emmanouil Vasilomanolakis, Jörg Daubert, Manisha Luthra, Vangelis Gazis, Alex Wiesmaier, and Panayotis Kikiras. 2015. On the security and privacy of Internet of Things architectures and systems. In *2015 International Workshop on Secure Internet of Things (SIoT)*. IEEE, 49–57.

[37] Ovidiu Vermesan, Peter Friess, et al. 2014. *Internet of things-from research and innovation to market deployment*. Vol. 29. River publishers Aalborg.

[38] ADDM4RIOTA Website. 2019. Retrieved March 7, 2019 from http://addm4riota.labnet.nce.ufrj.br