International Journal of Interactive Multimedia and Artificial Intelligence

March 2017, Vol IV, Number 3, ISSN: 1989-1660

IoT will stump IT until clouds and big data come aboard

Stephen Lawson, EMC (2016)

Special Issue on Advances and Applications in the IoT & Cloud Computing

INTERNATIONAL JOURNAL OF INTERACTIVE MULTIMEDIA AND ARTIFICIAL INTELLIGENCE
ISSN: 1989-1660–VOL. 4, NUMBER 3

Editor's Note

The Internet of Things is the networks of physical devices, embedded with electronics, software, sensors, actuators and security and connectivity mechanisms that enables them to collect and exchange data. It is a very important research topic nowadays in which many scientific papers are focusing on its bases [1].

This Special Issue tries to show some of the latest researches related to IoT with special emphasis on the basic components of IoT [2], some of the major applications in which researchers and practitioners are working [3] and especially in aspects related to security, one of the main areas of research related to IoT [4], with a special emphasis on cloud-based systems [5][6]. Next, I present a summary of the works that are included in this special issue.

Some of the key elements related to IoT are smart objects, sensors and actuators. So, González et al. present a review that explains the main concepts related to such elements, which can now be present in cities, houses, cars, through almost any physical item, capable of interconnecting with others in order to create a great range of opportunities. Authors also present one object classification system.

Wireless sensor networks are other determining factor for IoT. Bahuguna et al. show a study of the key factors that impacts the design and routing techniques of such networks. This is a very important topic since networks contains nodes with sensing, processing and communication capabilities, that have energy limitations as well as other requirements like connectivity and coverage.

IoT also opens the door to an unlimited number of applications. Dhall and Solanki present a use case on the automobile industry, using IoT-based technology and analytics. The goal is to provide a way to transmit information about the current status of vehicles and based on that information, create workflows to take actions related to car maintenance (e.g., scheduling a service with the manufacturer). The underlying idea is based on the concept of connected cars used to perform predictive car maintenance.

Continuing with the same topic, the collaboration between vehicles and the road side is very important to create intelligent transportation systems [7][8]. Vehicular Ad Hoc Networks (VANET) are important to provide comfort, safety and entertainment for people in vehicles. However, in order to give stable routes and adequate performance, there is a need of proper routing protocols. Rathi and Welekar, propose a routing protocol for such networks and evaluate its performance through a simulation.

In addition to the concept of connected cars, other authors such as Solanki et al., have also addressed issues related to smart cities. They present a method to preserve energy as well as water, in urban and rural areas through IoT. An autonomous system is proposed based on Arduino that is monitored with Lab View to control and interact with all the infrastructure located at various points in a city (parks, subways and highway lighting modules).

One extremely important factor in IoT is connectivity. Now we have a very different range of devices (both taking into account the hardware and the software), that are very difficult to communicate since they use different programming languages, protocols and interfaces. On that topic, Martínez et al., propose a migration process from classic C/C++ software applications to different mobile platforms. The proposal integrates standards with Haxe, a programming language that allows writing applications that target all major mobile platforms.

Cyber-physical attack attempts in IoT-based manufacturing systems are now very common. New threats to supply chain security have arisen, allowing attackers to manipulate physical features of pieces, resulting in more manufacturing costs. Pan et al., present two taxonomies: one for classifying cyber-physical attacks against manufacturing process and another for quality control measures for counteracting these attacks. They also provide a scheme for linking emerging vulnerabilities to possible attacks and quality control measures.

Since IoT is based, among other factors, on sensors, communication networks, data and processes that send information between devices, the protection of information traveling through them become very important. Gaona-García et al., present an analysis of previous works on security aspects related to the IoT, focusing at privacy levels and control access. They also provide a list of security issues that should ideally be addressed in these type of systems built with clusters.

Also related to security aspects, IoT relies on cloud computing to integrate and allow access to shared and configurable resources through the network. However, security is one of the biggest issues related to cloud-based and distributed systems. Thus, intrusion detection systems are required. Achbarou et al. present a classification of attacks against the availability, confidentiality and integrity of cloud resources and services, providing models to identify and prevent these types of attacks.

Given its proximity, and as stated in the previous work, security is a key aspect in both IoT and cloud computing. In Talbi and Haqiq, authors present a multi-agent based cloud service brokering system with the aim of analyzing and ranking different cloud providers, making decisions to know the more secured providers and justifying the business needs of users in terms of reliability and security.

Again with the idea of intrusion attacks in the huge amount of data that is generated in the IoT and the applications that are being moved to the cloud, Toumi et al. propose a collaborative framework between a hybrid intrusion detection system that is based on mobile agents and virtual firewalls. So, they propose three different layers to create a more reliable detection system.

To finish with proposals created to improve the cloud computing security, Saidi et al., show the most used techniques to avoid Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks in the cloud (e.g., HCF and CBF filters), which aims to break down the availability of a service to its legitimate users.

Dr. Vicente García-Díaz

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Comput. networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Futur. Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, 2013.

[3] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," Wirel. Pers. Commun., vol. 58, no. 1, pp. 49–69, 2011.

[4] R. H. Weber, "Internet of Things--New security and privacy challenges," Comput. Law Secur. Rev., vol. 26, no. 1, pp. 23–30, 2010.

[5] C. G. García, J. P. Espada, E. R. Núñez-Valdez, and V. García-Díaz, "Midgar: Domain-Specific Language to Generate Smart Objects for an Internet of Things Platform.," in IMIS, 2014, pp. 352–357.

[6] K. Venkateshwaran, A. Malviya, U. Dikshit, and S. Venkatesan,

DOI: 10.9781/ijimai.2017.430

"Security Framework for Agent-Based Cloud Computing.," Int. J. Interact. Multimed. Artif. Intell, vol. 3, no. 3, pp. 35–42, 2015.

[7] G. Cueva-Fernandez, J. P. Espada, V. García-Díaz, C. G. García, and N. Garcia-Fernandez, "Vitruvius: An expert system for vehicle sensor tracking and managing application generation," J. Netw. Comput. Appl., vol. 42, no. 0, pp. 178–188, 2014.

[8] G. C. Fernandez, J. P. Espada, V. G. Díaz, and M. G. Rodríguez, "Kuruma: the vehicle automatic data capture for urban computing collaborative systems," Int. J. Interact. Multimed. Artif. Intell., vol. 2, no. 2, pp. 28–32, 2013.

TABLE OF CONTENTS

EDITOR'S NOTE IV

A REVIEW ABOUT SMART OBJECTS, SENSORS, AND ACTUATORS 7

AN ANALYTIC STUDY OF THE KEY FACTORS INFLUENCING THE DESIGN AND ROUTING TECHNIQUES OF A WIRELESS SENSOR NETWORK 11

AN IOT BASED PREDICTIVE CONNECTED CAR MAINTENANCE APPROACH 16

PERFORMANCE EVALUATION OF AODV ROUTING PROTOCOL IN VANET WITH NS2 23

CONCEPTUAL MODEL FOR SMART CITIES: IRRIGATION AND HIGHWAY LAMPS USING IOT 28

MIGRATING C/C++ SOFTWARE TO MOBILE PLATFORMS IN THE ADM CONTEXT 34

TAXONOMIES FOR REASONING ABOUT CYBER-PHYSICAL ATTACKS IN IOT-BASED MANUFACTURING SYSTEMS 45

ANALYSIS OF SECURITY MECHANISMS BASED ON CLUSTERS IOT ENVIRONMENTS 55

SECURING CLOUD COMPUTING FROM DIFFERENT ATTACKS USING INTRUSION DETECTION SYSTEMS 61

A MAS-BASED CLOUD SERVICE BROKERING SYSTEM TO RESPOND SECURITY NEEDS OF CLOUD CUSTOMERS 65

USE TRUST MANAGEMENT FRAMEWORK TO ACHIEVE EFFECTIVE SECURITY MECHANISMS IN CLOUD ENVIRONMENT 70

TECHNIQUES TO DETECT DOS AND DDOS ATTACKS AND AN INTRODUCTION OF A MOBILE AGENT SYSTEM TO ENHANCE IT IN CLOUD COMPUTING 75

OPEN ACCESS JOURNAL

ISSN: 1989-1660

COPYRIGHT NOTICE

A review about Smart Objects, Sensors, and Actuators

Cristian González García, Daniel Meana-Llorián, B. Cristina Pelayo G-Bustelo, and Juan Manuel Cueva Lovelle

University of Oviedo, Department of Computer Science, Oviedo, Spain

Abstract — **Smart Objects and the Internet of Things are two ideas which describe the future, walk together, and complement each other. Thus, the interconnection among objects can make them more intelligent or expand their intelligence to unsuspected limits. This could be achieved with a new network that interconnects each object around the world. However, to achieve this goal, the objects need a network that supports heterogeneous and ubiquitous objects, a network where exists more traffic among objects than among humans, but supporting for both types. For these reasons, both concepts are very close. Cities, houses, cars, machines, or any another object that can sense, respond, work, or make easier the lives of their owner. This is a part of the future, an immediate future. Notwithstanding, first of all, there are to resolve a series of problems. The most important problem is the heterogeneity of objects. This article is going to show a theoretical frame and the related work about Smart Object. The article will explain what are Smart Objects, doing emphasis in their difference with Not-Smart Objects. After, we will present one of the different object classification system, in our opinion, the most complete.**

Keywords — **Smart Objects, Internet of Things, Sensors, Actuators.**

I. What is an Object?

Throughout the difference literature about the universe of the Internet of Things, we could see the word ‘object’ in general. Why? The reason is simple. The word ‘object’ is used to refer to any device or thing, which can be intelligent or not. It means that when people talk about the interconnection among objects they talk about the interconnection among Smart Objects, among Not-Smart Objects, or between both.

Nevertheless, there are a lot of problems in the literature and the people’s understanding when the word ‘object’ is said, furthermore, some people use the word ‘thing’ instead of ‘object’, or some authors use both words interchangeably. The reason is that both definitions are very ambiguous due to the use of these word in the articles or our daily life. This is why, in this first section, we are going to introduce the exact meaning in order to delete this ambiguity.

Object according to WordReference [1]

1. Anything that can be seen or touched and is for the most part stable or lasting in form, and is usually not alive.

2. A thing, person, or matter to which thought or action is directed; the cause of such thought or action.

Thing according to WordReference [1]

1. An object, usually not a person or animal.

Object according to Oxford [2]

1. A material thing that can be seen and touched.

Thing according to Oxford [2]

1. An object that one need not, cannot, or does not wish to give a specific name.

Object according to Cambridge [3]

1. A thing that can be seen or felt.

Thing according to Cambridge [3]

1. An object; something that is not living.

As we could see, the definition depends on the site where you consult and, even so, in these cases, the definitions can be very ambiguous. The definition that is better adapted for the typical use of the word ‘object’ in the universe of the Internet of Things is the first definition of ‘object’, the definition obtained from WordReference. Based on this, one possible definition to the word ‘object’ in the universe of the Internet of Things could be:

Any electronic device that can be connected to the Internet and collect data, like a sensor, or perform an action in an object, normally called actuator.

In the following sections, we will detail the differences between Smart Objects and Not-Smart Objects, explaining what are their and using examples of each one.

II. Not-Smart Objects

In the previous section, we explained what the **objects** are and what elements compose this group. These elements are the objects without intelligence and objects with intelligence which are also known as **Smart Objects**. Due to the existence of these elements, it is essential to know how to distinguish the different type of objects and know the way in which these objects can interact with us. In this section we will address the objects of the second group, the objects without intelligence or Not-Smart Objects, and in later sections we will deepen in the **Smart Objects**. The Not-Smart Objects can be formed by **sensors** and **actuators**.

Sensors are electronic devices composed of sensitive cells [1] that are able to measure physical parameters like the light fluctuation using a photoresistor, the temperature using a thermistor, to detect flames, sounds, movements, or any other fluctuation in the environment [1], [4]. Thus, sensors are specific physical elements that allow us to

DOI: 10.9781/ijimai.2017.431

measure a concrete physical parameter or detect something of the sensor's immediate environment.

However, **actuators** can be **mechanic actuators** which allow actions over themselves or over other devices, and **actions** which a specific object allow to perform. Thus, we can divide **actuators** in two different groups: mechanic devices and actions. Examples of **mechanic actuators** could be motors, servomotors or hydraulic bombs, and examples of **actions** could be to send a message, control LEDs, turn on lights or control the movement of a robot or any other available robot's actions.

According to the previous definitions, we could find devices that combine both types of Not-Smart Objects, they not only would have actuators and sensors but also would have both. An example of these are smartphones or any other Smart Object that are composed by sensors and actuators. Another similar example could be a microcontroller like an Arduino. The Arduino microcontroller is capable of manage almost any type of electronic device. Thus, an Arduino allows creating a system composed only of actuators, only of sensors, or both. Therefore, the **Smart Objects** are formed by Not-Smart Objects.

Fig. 1 shows a concept map that explains the composition of the **objects**. This Fig. is useful to understand better the difference between Not-Smart Objects and **Smart Objects**. As we can see in Fig. 1, Not-Smart Objects can be sensors or actuators, and actuators are divided into mechanic actuators and actions. Moreover, in order to improve the understandability, Fig. 1 shows several examples of each group.

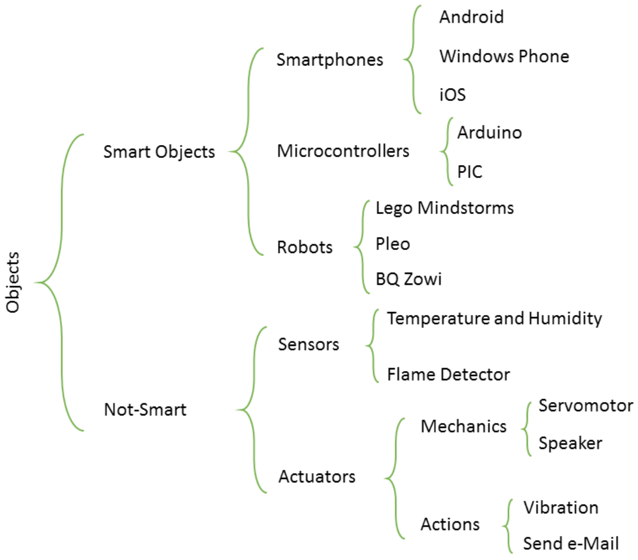


Fig. 1 Composition of objects using examples

III. Smart Objects

The definition of **Smart Object** depends on its author. Nevertheless, some authors agree with other authors and therefore, we can get a premise of their definitions. Below is our premise which was created using the definitions obtained from [5]–[9].

A Smart Object, also known as **Intelligent Product**, is a physical element that can be identified throughout its life and interact with the environment and other objects. Moreover, it can act in an intelligent way and independently under certain conditions. Furthermore, **Smart Objects** have an embedded operating system and they usually can have actuators, sensors, or both [5]. This allows Smart Objects to communicate with other objects, process environment data, and do events. However, there are definitions that differ from the previous which was obtained from [5]–[9].

The definition from [10] is very different from the previous. In [10], they consider as **Intelligent Products** the objects which are constantly monitoring, which react and adapt to the environment, which have an optimum performance, and which hold an active communication.

In our daily life, we are surrounded by examples of **Smart Object** and there are also examples in our everyday objects like smartphones, tablets, Smart TVs, microcontrollers like Arduino [5], [11]–[13], and even some coffee pots and some cars are also Smart Objects. Therefore, an object connected to the Internet [14] and capable of manage information [15] can be a **Smart Object.**

As we can see, Smart Objects can be very different from each other. A smartphone has little in common with a microcontroller and microcomputer. They only have in common some electronic components. Each one has their own sensors and actuators, their own intelligence, and their own operating system when they have one.

Smart Objects can be classified through three dimensions according to [15] and like Fig. 2 shows. This classification is useful to distinguish the different data that a Smart Object can give us about its architecture. Each dimension represents a quality of the intelligence. With the three dimensions, we can determine the intelligence that an object has and the type of **Smart Object** that it is. The three dimensions are the level of intelligence, the location of the intelligence, and the aggregation level of the intelligence.

A. *Level of intelligence*

The first dimension is the level of intelligence. This describes how much intelligent an object can be. It is formed by three levels **information handling**, **notification of the problem,** and **decision making**.

1) *Information handling*

The **information handling** is the capacity of the object to manage the information gathered from sensors, readers, or from any other techniques.

This is the most basic intelligence level and all Smart Object must have it, thus, any Smart Object must be able to manage the information that receives. Otherwise, it would not be a Smart Object and it would be just a **Not-Smart Object**.

2) *Notification of the problem*

The **notification of the problem** is the ability of an object to notify its owner under certain conditions or when an event occurs like flames detection, an unusual decrease of the temperature, or any other event like these. In this level, the **objects** do not have free will.

3) *Decision making*

The **decision making** is the highest level of intelligence that an object can have. An **object** has this level when it has the other two levels and it is able to take decisions by itself. It does not require any type of intervention, thus, it has free will.

B. *Location of the intelligence*

The second dimension is the location of the intelligence and is formed by two categories according to [15], but we have added one extra category. Thus, this dimension has three categories: **intelligence through the Network**, **intelligence in the Object**, and **combined intelligence**. Moreover, we added a third level that combines both levels.

1) *Intelligence through the Network*

The **intelligence through the Network** consists in that the intelligence depends totally on an external agent due to the lack of intelligence in the object. This agent can be a network where the object is connected, usually known as **portal platforms** [16], a server that runs the agents or another object that takes decisions or has the global intelligence.

2) *Intelligence in the Object*

The **intelligence in the Object** means that the objects with this level, can process information by themselves, so, they do not need any external agent in order to be intelligent. The platforms that have **objects** with this level are usually called **embedded platforms** [16].

3) *Combined intelligence*

The **combined intelligence** is a level that [15] does not include in their classification but they talk about it and they include it in an example graph. In this level, the object has the both intelligences. It has its own intelligence and it is capable of use the intelligence located in the Network. This platforms are usually called **surrogated platforms** [16].

C. *Aggregation level of the intelligence*

The last dimension is the **aggregation level of the intelligence** which is formed by three categories. This dimension is useful to describe the **objects** that are composed of several parts. Depending on the aggregation level we could say that an object is indivisible or every part is independent. For example, we can connect a Raspberry Pi with an Arduino and connect sensors or actuators to both devices. The Not-Smart Objects like the sensors or actuators, do not have their own intelligence but the Raspberry Pi and the Arduino are Smart Objects. Therefore, if we disconnected the Arduino and the Raspberry Pi, they could run independently, whereas if we disconnect the Not-Smart Objects they could not work by themselves.

The two categories are: **intelligence in the item**, **intelligence in the container**, and **distributed intelligence**.

1) *Intelligence in the item*

The first category is the **intelligence in the item**. This category includes the objects that are capable of handling information, notifications and/or decisions. Moreover, if these objects are composed of different components, these components must not be independents. Examples of objects that belong to this category are the smartphones. They are composed of sensors and actuators that cannot be separated because they are embedded.

2) *Intelligence in the container*

The second category is the **intelligence in the container**. The objects of this category must be able to handle information, notifications and/or decisions and they must know their components in order to work as a proxy between their components and the Internet or the intelligence. Moreover, these **objects** are capable of working as containers or Smart Objects in spite of removing some of their components. An Arduino with at least two sensors belongs to this category. If we removed a sensor from the Arduino, the Arduino would be able to continue working as container. Another example could be intelligent shelve [15] that notify when a product is out of stock.

3) *Distributed Intelligence*

The second category is the **distributed intelligence**. This category is the fusion between the other two. Here, items and containers have intelligent but, in this case, they can negotiate between themselves according to take the best decision to the object in base on the whole system and the rest of items. This category was added by us because we have worked with object which need this interaction. An example of this category is when you have a Smart Object which is composed by other Smart Objects, for instance, a Raspberry Pi which has connected two Arduinos. In this case, each Arduino has its own intelligence and it can take their own decisions, but sometimes, it has to ask to the Raspberry Pi about some data or the state of the another Arduino to do some action.

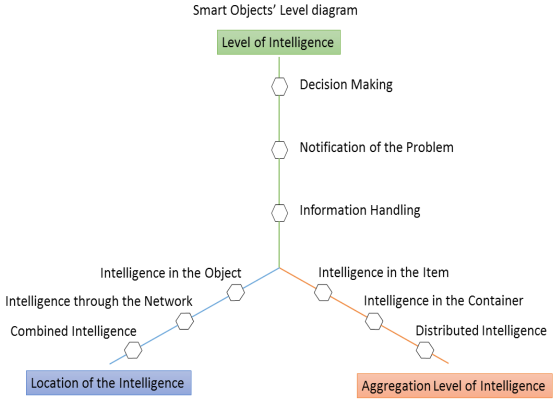

Smart Objects' Level diagram
Level of Intelligence
Decision Making
Notification of the Problem
Information Handling
Intelligence in the Object
Intelligence through the Network
Combined Intelligence
Location of the Intelligence
Intelligence in the Item
Intelligence in the Container
Distributed Intelligence
Aggregation Level of Intelligence

Fig. 2 Classification of the intelligence based on Meyer's classification

From our point of view, the most important type of **Smart Objects** is the **combined intelligence** or the **intelligence through the network** from the dimension **location of the intelligence**. This type of objects alongside with the Internet of Things, allow adding intelligence to the network and actions according to the data that these objects collect, and services that they offer. In this way, the objects that are connected to the network could have intelligence, or even, be more intelligent.

IV. Application Areas

Smart Objects are presents in our daily life for a long time. We usually consider that **Smart Objects** and the **Internet of Things** go together, although, there are many examples about the usage of **Smart Objects** without the usage of the **IoT**.

We can find Smart Objects in different systems on the **commercial field** in order to control the manufacturing like happens in [6]. Furthermore, in [7], [15], they use **Smart Objects** in order to improve the distribution and the products management in supply chains to have the products located during all their life cycle.

In the first paper, the authors describe when use different elements like readers in order to know the states of the products, monitoring them, and access to their history. Whereas, in the second paper, they mention an example which we already talked about. They mention intelligent shelves which notify when a product is out of stock.

These kind of applications are very useful to companies because they obtain advantages to improve and avoid problems related with the lack of stock during the all chain of the product life.

Following with possible uses of Smart Objects, another usage is proposed in [17]. This proposal consists in analysing the usage of rented items in order to collect the appropriate quantity of money, and also, punishing improper usage of the object. The system is good for clients as well as for companies. The clients pay exactly for the usage of the product and companies can detect improper usages and be compensated.

Smart Objects can also be used to improve the safety at work. In [17], the authors proposed a system to alert nearby employees about the incorrect and insecure storage of chemical material. The system proposed can be very useful because it allow managing the storage of hazardous substances and avoiding many problems or disasters.

The medical field is another field where Smart Objects can be used. A research in this field is [18]. In this research, the authors proposed a system that monitor patients with problems. Thanks to systems like this, many human lives could be saved. An example could be the connection of a cardiac pacemaker with a monitoring centre in order to detect, immediately, heart attacks or failures in the pacemaker.

V. Conclusions

In this paper, we analysed the differences between Smart objects and Not-Smart Objects. In the literature, we cannot find the exactly differences or we can see as some authors use the both words indistinctly. In fact, this creates a problem to understand the exactly devices that they use.

Smart Objects can be used for resolving a lot of problems. We have showed a few examples of it, from supply chain to security and health.

Notwithstanding, as we say before, objects need a central system to create the interconnection between themselves. According to this goal, we can use a specific system or some Internet of Thing (IoT) platform. We can see some examples in [19] and a classification of the different IoT network types in [20]. The last one contains examples of different IoT platforms which support heterogeneous and ubiquitous objects and interconnect the objects between themselves.

We can see that the combination between Smart Objects and the Internet of Things can offer many advantages and improve the peoples' life because it can interconnect and communicate the different object to create more complex applications. Besides, we have added two new categories to the Meyer's classification in order to adapt to the new type of objects and applications in the Internet of Things.

Acknowledgment

This work was performed by the 'Ingeniería Dirigida por Modelos MDERG' research group at the University of Oviedo under Contract No. FC-15-GRUPIN14-084 of the research project 'Ingeniería Dirigida Por Modelos MDERG'. Project financed by PR Proyecto Plan Regional.

References

[1] WordReference.com LLC, "WordReference," 2016. [Online]. Available: http://www.wordreference.com/. [Accessed: 17-Feb-2016].

[2] Oxford University Press, "Oxford Dictionaries," 2016. [Online]. Available: http://www.oxforddictionaries.com/. [Accessed: 18-Feb-2016].

[3] Cambridge University Press, "Cambridge Dictionaries Online," 2016. [Online]. Available: http://dictionary.cambridge.org/. [Accessed: 18-Feb-2016].

[4] Alphabet Inc., "Google," 2016. [Online]. Available: http://www.google.es. [Accessed: 17-Feb-2016].

[5] K. A. Hribernik, Z. Ghrairi, C. Hans, and K. Thoben, "Co-creating the Internet of Things - First Experiences in the Participatory Design of Intelligent Products with Arduino," in *Concurrent Enterprising (ICE), 2011 17th International Conference on*, 2011, pp. 1–9.

[6] D. McFarlane, S. Sarma, J. L. Chirn, C. . Wong, and K. Ashton, "Auto ID systems and intelligent manufacturing control," *Eng. Appl. Artif. Intell.*, vol. 16, no. 4, pp. 365–376, Jun. 2003.

[7] C. Y. Wong, D. McFarlane, A. Ahmad Zaharudin, and V. Agarwal, "The intelligent product driven supply chain," in *IEEE International Conference on Systems, Man and Cybernetics*, 2002, vol. vol.4, p. 6.

[8] R. Van Kranenburg, D. Caprio, E. Anzelmo, A. Bassi, S. Dodson, and M. Ratto, "The Internet of Things," in *1st Berlin Symposium on Internet and Society*, 2011, no. October 2015, p. 84.

[9] M. Kärkkäinen, J. Holmström, K. Främling, and K. Artto, "Intelligent products—a step towards a more effective project delivery chain," *Comput. Ind.*, vol. 50, pp. 141–151, 2003.

[10] O. Ventä, *Intelligent products and systems: Technology theme-final report*, no. 635. VTT Technical Research Centre of Finland, 2007.

[11] V. Georgitzikis, O. Akribopoulos, and I. Chatzigiannakis, "Controlling Physical Objects via the Internet using the Arduino Platform over 802.15.4 Networks," in *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 2012, vol. 10, no. 3, pp. 1686–1689.

[12] A. Piras, D. Carboni, and A. Pintus, "A Platform to Collect, Manage and Share Heterogeneous Sensor Data," in *Networked Sensing Systems (INSS)*, 2012, pp. 1–2.

[13] Arduino, "Arduino," 2016. [Online]. Available: https://www.arduino.cc/. [Accessed: 09-Feb-2016].

[14] G. M. Lee and J. Y. Kim, "Ubiquitous networking application: Energy saving using smart objects in a home," in *2012 International Conference on ICT Convergence (ICTC)*, 2012, pp. 299–300.

[15] G. G. Meyer, K. Främling, and J. Holmström, "Intelligent Products: A survey," *Comput. Ind.*, vol. 60, no. 3, pp. 137–148, Apr. 2009.

[16] F. Ramparany and O. Boissier, "Smart devices embedding multi-agent technologies for a pro-active world," in *Proc. Uniquitous Computing Workshop*, 2002.

[17] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the Internet of things," *IEEE Internet Comput.*, vol. 14, no. 1, pp. 44–51, Jan. 2010.

[18] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[19] C. G. García, C. P. García-Bustelo, J. P. Espada, and G. Cueva-Fernandez, "Midgar: Generation of heterogeneous objects interconnecting applications. A Domain Specific Language proposal for Internet of Things scenarios," *Comput. Networks*, vol. 64, no. C, pp. 143–158, Feb. 2014.

[20] C. G. García, J. P. Espada, E. R. N. Valdez, and V. G. Diaz, "Midgar: Domain-Specific Language to Generate Smart Objects for an Internet of Things Platform," in *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2014, pp. 352–357.

Cristian González García is a Technical Engineering in Computer Systems and M.S. in Web Engineering from School of Computer Engineering of Oviedo in 2011 and 2013 (University of Oviedo, Spain). Currently, he is a Ph.D. candidate in Computers Science. His research interests are in the field of the Internet of Things, Web Engineering, Mobile Devices, and Modelling Software with DSL and MDE.

Daniel Meana-Llorián is a Graduated Engineering in Computer Systems from School of Computer engineering of Oviedo in 2014 (University of Oviedo, Spain). Currently, he is a student of M.S. in Web Engineering. His research interests include Mobile technologies, Web Engineering, the Internet of Things, and exploration of emerging technologies related with the previous ones.

B. Cristina Pelayo G-Bustelo is a Lecturer in the Computer Science Department of the University of Oviedo. Ph.D. from the University of Oviedo in Computer Engineering. Her research interests include Object-Oriented technology, Web Engineering, eGovernment, and Modelling Software with BPM, DSL, and MDA.

Juan Manuel Cueva Lovelle is a Mining Engineer from Oviedo Mining Engineers Technical School in 1983 (Oviedo University, Spain). Ph. D. from Madrid Polytechnic University, Spain (1990). From 1985 he is Professor at the Languages and Computers Systems Area in Oviedo University (Spain). ACM and IEEE voting member. His research interests include Object-Oriented technology, Language Processors, Human-Computer Interface, Web Engineering, and Modelling Software with BPM, DSL, and MDA.

An analytic Study of the Key Factors Influencing the Design and Routing Techniques of a Wireless Sensor Network

Yogita Bahuguna, Deepak Punetha, Pooja Verma

Department of Electronics & Communication Engineering, Tula's Institute, Dehradun, India

***Abstract* — A wireless sensor network contains various nodes having certain sensing, processing & communication capabilities. Actually they are multifunctional battery operated nodes called motes. These motes are small in size & battery constrained. They are operated by a power source. A wireless sensor network consists of a huge number of tiny sensor nodes which are deployed either randomly or according to some predefined distribution. The sensors nodes in a sensor network are cooperative among themselves having self-organizing ability. This ensures that a wireless network serves a wide variety of applications. Few of them are weather monitoring, health, security & military etc. As their applications are wide, this requires that sensors in a sensor network must play their role very efficiently. But, as discussed above, the sensor nodes have energy limitation. This limitation leads failure of nodes after certain round of communication. So, a sensor network suffers with sensors having energy limitations. Beside this, sensor nodes in a sensor network must fulfill connectivity & coverage requirements. In this paper, we have discussed various issues affecting the design of a wireless sensor network. This provides the readers various research issues in designing a wireless sensor network.**

***Keywords* — Wireless Sensor Network, Nodes Energy, QoS, Connectivity.**

I. Wireless Sensor Network

With the advancement of technology the area of applications of wireless sensor network (WSN) are growing day by day. A WSN contains many sensor nodes of the order of hundreds or even thousands. They are tiny in size having limited energy and communication range. Besides this, they suffer from limited sensing, computational and transmission capabilities. In a wireless network information is transferred from a source (sensor nodes) to a sink (base station). The base station can be placed inside or outside the monitoring area. The sensor nodes are scattered over a huge geographical area randomly [1-3] [14]. The nodes in a wireless network can either communicate among themselves or to a base station. These nodes are sophisticated having intra and inter communication capabilities. In a wireless network the nodes play different roles as per the necessity of communication. A node can provide an interfacing between sources and sink thereby reducing the energy consumption by providing an energy efficient path to route the data from the source to the sink [11]. These nodes are called gateways. Gateway nodes selection is essential to prevent the death of the nodes due to excessive energy consumption. Relay nodes (routers) are used to expand the coverage area and to provide backup routes in case of failure of nodes and data traffic.

A leaf node (endpoint) is used to establish an interfacing between a wireless network and a sensor that is wired to it. Actuators are the sensors used to interact with the physical environment which has to be controlled. The vital applications of wireless networks are enormous. This section depicts various areas of application of WSNs [8]. They serve the different purpose starting from home applications, health applications and military applications to the environmental applications. In the environmental applications they are used to monitor a wide range of ambient conditions such as temperature, humidity, pressure, noise level and soil make up etc which affect the crops and live stocks [24]. Moreover, in forest fire and flood detection, earth monitoring and pollution study the use of wireless network is crucial. The wireless networks are being used in health sector efficiently in various fields. To monitor doctor and patient inside a hospital, to monitor the activities in a hospital, to monitor the activities and processes in tiny insects are few of them [17]. Beside these, sensors can be deployed in a health care unit to control drugs administration, thus ensuring the safe and the better drug services to the patients. WSNs are playing an unavoidable role in home applications as well. Several home appliances such as microwave ovens, vacuum cleaners, refrigerators are now equipped with sensors, thus enabling these appliances to interact with one another or to an external link via the internet and finally offering better services to the

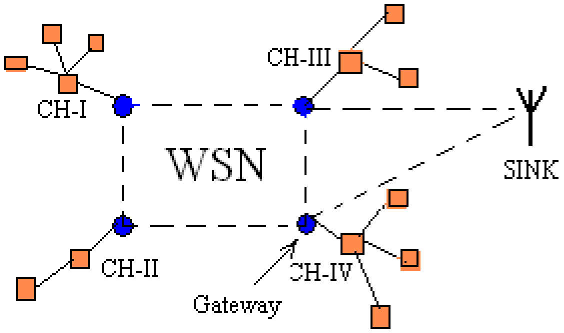


Fig. 1. A basic block diagram of a WSN

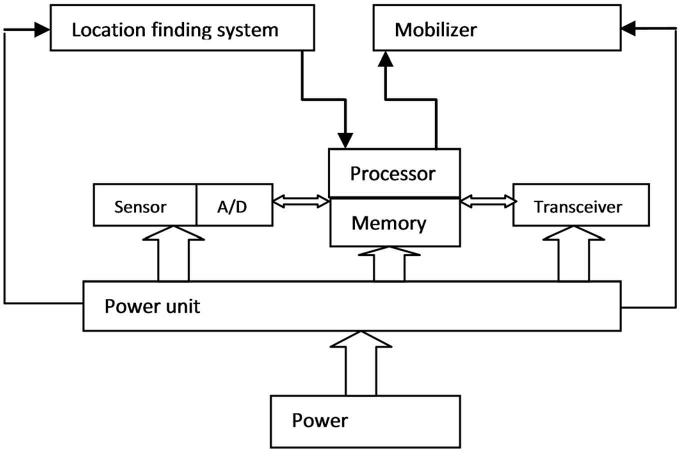


Fig. 2. A block diagram of a WSN.

DOI: 10.9781/ijimai.2017.432

end user [10] [16]. Last but not the least, the role of WSNs in military application is pervasive. Sensor nodes are deployed in remote places where human interference except military is not possible [23]. They are used to monitor intrusion of terrorists, equipment, vehicle etc from the outside. Wireless sensors can be used to gather data of battlefield [12] [22] [25]. In all the applications the role of sensor node is crucial. Therefore, remedies have to take in order to minimize decay of nodes, so as to prolong the network lifetime.

II. Network Structure

A sensor node comprises of four main units namely a sensing unit, a processing unit, a transceiver unit and a power unit (fig.1). The sensing unit can be further divided into two parts: sensors and an analog to digital converter [13] [15]. The function of the sensor is to first sense the physical phenomenon.

Then it converts the sensed physical phenomenon into an analog signal. The output of the sensor is then fed to an analog to digital converter circuitry which converts an analog signal into digital signal appropriate for further transmission. The output of the sensing unit is then applied to the processing unit which generally consists of a processor and a storage device. The function of a processor is to generate necessary instructions and commands in order to facilitate the coordination among the sensors node [4]. There by performing the sensing task. The third unit is transceiver which may be a passive or active optical device [20]. The term transceiver is used for the transmitter and receiver combined, which is equipped with the ability of transmitting and receiving a signal. The transceiver unit of sensor nodes connects the node to the network. The fourth unit is a power unit driven by a power generator such as a solar cell which uses energy scavenging techniques and extracts energy from the environment. Energy scavenging is very important since sensor nodes may remain un- attendant in a remote place for several months or years without human interference [18]. Beside these four, two sub-units are a location finding system and a mobilizer. The location finding system is necessary in a WSN to apply routing techniques in a proper way. The routing techniques use the information about the sensor's location provided by the location finding system. A mobilize is optional which is needed in case of mobile sensor nodes.

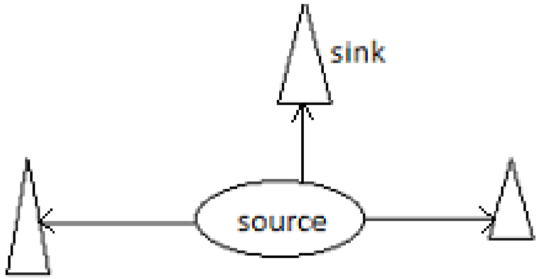


Fig. 3.1 Single Hop Communication.

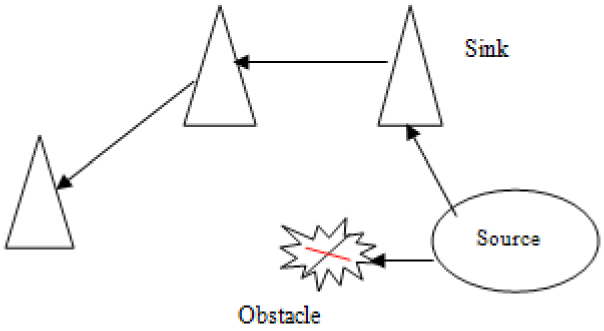


Fig. 3.2 Multi Hop Communication

A WSN can be of two types depending upon the manner of data transmission viz. single hop and multi hop WSN. In case of single hop transmission (fig.2.1) data is transferred from gateway to the sensor node whereas in case of multi hop communication (fig.2.2) two or more sensor nodes are used in data path. In multi hop network, the data is transferred from one node to another in case obstacle like hills, mountains, lakes etc is found in the data path. Many routing techniques are based on multi hop networking of WSNs.

III. Problem Definition

As discussed in the earlier sections that sensor nodes are energy constraints operated by power source like solar cells. These sensor nodes are deployed randomly or according to some predefined criterion. But in general deployment of sensor nodes can be done randomly over a region of interest by dropping them from the helicopter or manually [21]. These nodes have self-organizing capability and they gather the information from the event under consideration, transmit the collected information to the sink via neighbor nodes. They are used in several applications like health monitoring, environment monitoring, home monitoring and military applications where the role of sensor node is most important. But sensor nodes are limited in terms of energy and range of communication, therefore careful deployment strategies and design parameters have to be considered when using WSNs in the above applications. In the past years many factors that affect the design of WSNs have been elaborated and different routing techniques based on minimizing the power utilization by the sensor nodes came into existence. In this section we give an overview of the issues that influence the design of WSNs and that must be overcome in order to facilitate effective communication.

A. Nodes deployment

The manner in which deployment of sensor nodes is done greatly affects the design of sensor network as well as routing techniques. As discussed in the previous sections that the deployment of sensor nodes could be done either in a random fashion or according to predefined set of rules (fig.3). Whatever is the way of deployment, the number of sensor nodes should be well enough to cover the whole geographical area. The density of sensor nodes in a geographical area of 1m3 can be as high as 20 nodes [19]. However the distribution of sensor nodes depends on the need of application. In

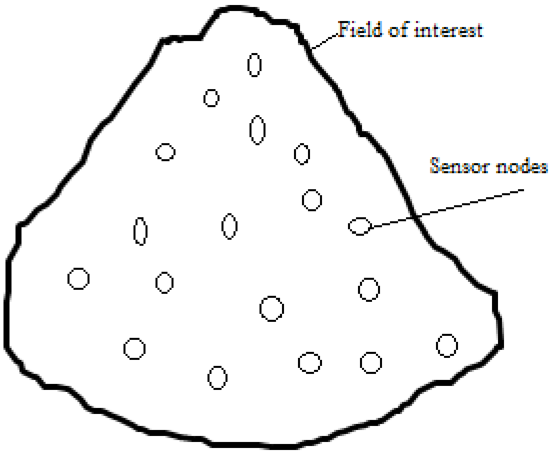


Fig. 4. Nodes Deployed in a Field of Interest.

the deployment phase, the topology that minimizes the installation cost while maximizing the flexibility should be adopted. Failure of the sensor nodes is common in the network; therefore the need of introducing the new sensors arises in the existing network. The deployment topology used should be flexible to manage with the introduction of new sensor nodes. Beside this, the topology adopted should be able to prop up the self-organizing quality of the sensor node in a network.

B. *Power consumption*

In WSNs many routing protocols have been proposed. They mainly consider the power requirements of a sensor network. This implies energy limitations of a sensor network have great influence on the routing techniques. The sensor nodes are multifunctional; they work as a sensor node and as gateways in the data path [4-5] [9]. Therefore for a WSN to work properly it becomes important that its sensor can conserve their energy so that network lifetime can be prolonged. If by chance a sensor failure occurs due to energy loss, it should not have an effect on the whole WSN. Usually multipath routing is adopted to transmit the data from the source to the sink (base station). Although single hop routing can also be applied to a WSN at the cost of more power consumption.

C. *Mobility of sensor node*

Sensor nodes are one of the most important components of a wireless sensor network [6]. A sensor node plays multifunctional roles viz. a data router or a gateway. Many applications may require static behavior of sensor nodes. Whereas, some application may need sensor nodes to behave dynamically. This necessitates the design of wireless network to be able to facilitate the mobility of sensor node.

D. *Fault tolerance*

The sensor network consists of a huge number of tiny sensors which are prone to frequent failures due to various reasons. Since they are battery operated, therefore insufficiency of power is one of the main causes of nodes death [5] [11]. Other reasons may include disturbances from the environment and physical damage. Due to mentioned faults if any of the nodes fail, it should not influence the overall functioning of the sensor network. Therefore, a WSN should be well reliable or trust worthy to cope up with the disturbances arising due to nodes death. Thereby smoothening the overall performance of the sensor network and sustaining the networks life. The Poisson's distribution provides the mathematical expression for the reliability of a sensor node K of not having a failure within a time interval (0, t) ;

$$Rk\ (t) = \exp(-\lambda k\ t) \tag{1}$$

Where R_k (t) is the reliability of a K^{th} sensor node, λ_k is the failure rate of sensor node K and t is the time period

E. *Security*

With the advancement of new technologies the areas of application of WSNs have grown surprisingly [6]. There use in mission critical operations such as battlefield surveillance, intrusion detection and target monitoring etc. creates the concern on security issues. In case of a mobile sink or gateways security is a crucial issue to be considered when designing wireless sensor network and routing protocol. Moreover, the wireless nature of the network, dense deployment of nodes, resource limitations of nodes and unavailability of fixed infrastructure creates a big concern on the network security. The wireless networks are more prone to attacks than wired networks. The design of a wireless network should be flexible in the sense that security level is changed when there is a change in available resources.

F. *Communication energy*

The main objective of a wireless sensor network is data communication which involves both data transmission and data reception [7]. Data is transferred from the sensor node to the base station via some gateways. As the number of data transmissions is increased, more energy will be devoted in the communication process.

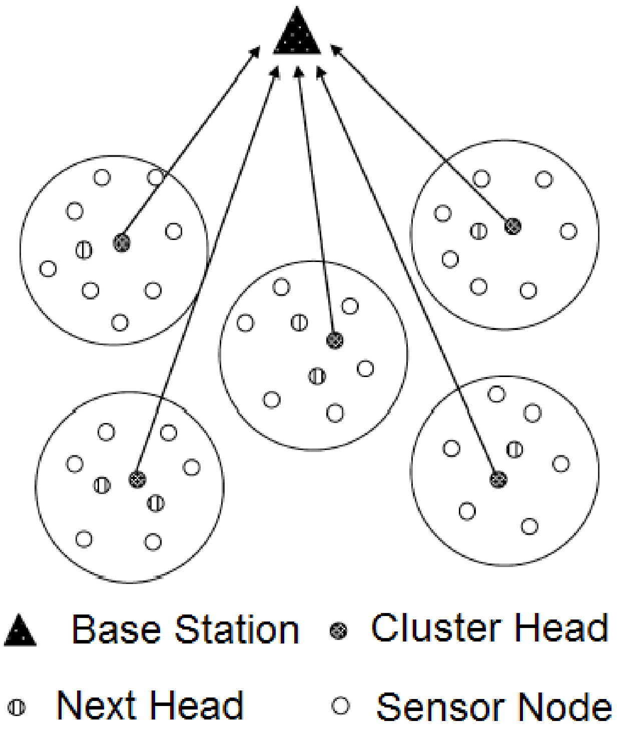


Fig. 5. Cluster formation.

Therefore to lessen the number of transmissions from the base station to the sink nodes, clustering is viable.

IV. Simulation and Experiment Results

The simulation result has been done by MATLAB tool. ECSSCoM (Energy-aware Clustering Sensor Scheduling Coverage Maintenance) a coverage maintenance protocol for wireless sensor networks. This protocol uses two techniques namely network clustering and sensor

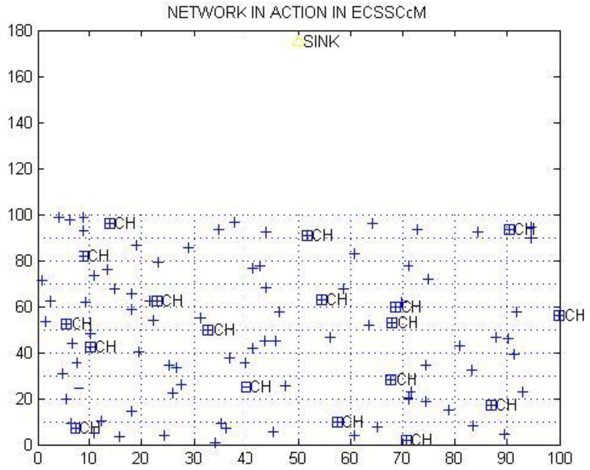


Fig. 6.1 Node deployment in ECSSCoM.

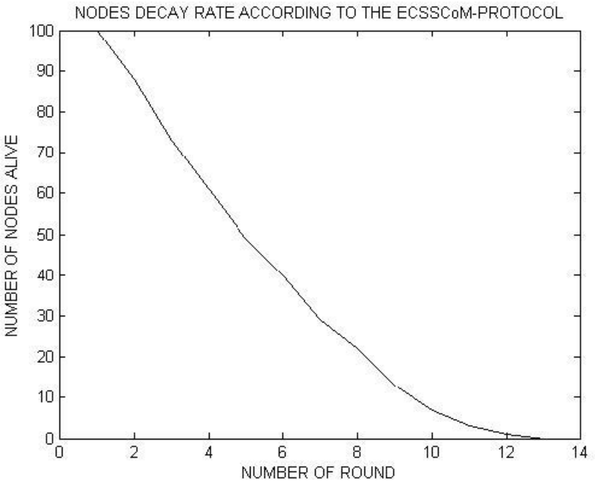


Fig. 6.2 Performance of nodes in ECSSCoM.

activity scheduling.

The simulation has been categories in several platform in which simulation of ECSSCoM has been performed as well as simulation of ECSSCoM and improved ECSSCoM with object consideration taken place.

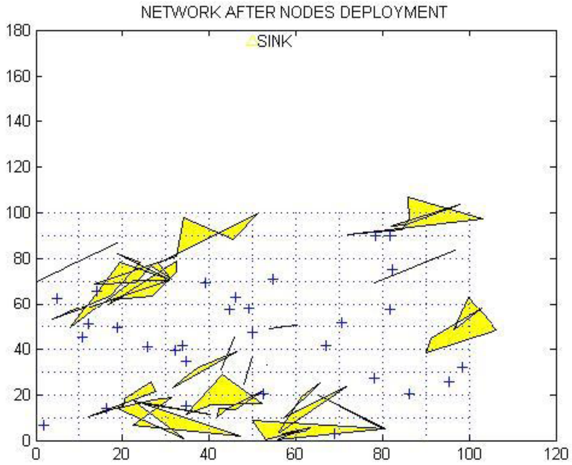


Fig. 7.1 Node deployment in ECSSCoM with obstacle consideration

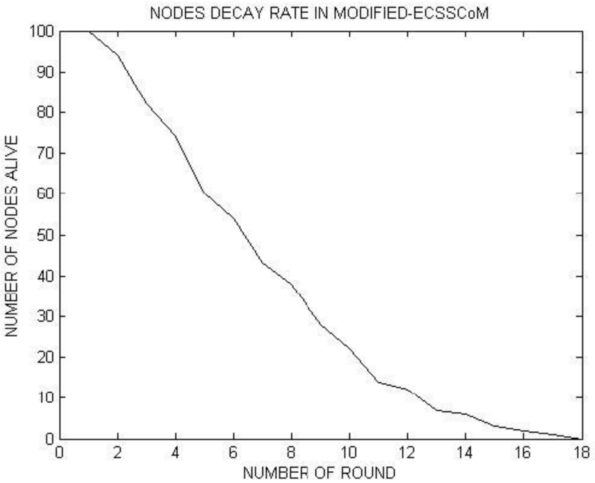


Fig. 7.2 Performance of nodes in ECSSCoM with obstacle.

The analysis and simulation results have been obtained for ECSSCoM for 100 nodes with energy of 2 Joule. In this execution it has been observed that every nodes of the WSN first find the neighboring nodes after that it form a cluster head.

V. Conclusion

In this paper, a comprehensive study has been executed over various key parameter like nodes deployment, mobility of sensor nodes, communication energy and power consumption etc. The parameters discussed in this manuscript play an imperative role in designing the wireless sensor network simultaneously providing a platform for new research areas on designing and routing techniques for WSNs by pinpointing the various challenges.

References

[1] V.B. Semwal, V.B. Semwal, M. Sati and S. Verma, “Accurate location estimation of moving object in Wireless Sensor network,” International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 1, No. 4 , pp. 71-75, 2011.

[2] García, Óscar, Ricardo S. Alonso, Dante I. Tapia, and Fabio Guevara, “Wireless Sensor Networks and Real-Time Locating Systems to Fight against Maritime Piracy,” International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 1, no. 5, pp. 14-21, 2012.

[3] Yogita Bahuguna, Jyoti Rawat, “An Efficient Routing Protocol in Heterogeneous Wireless Sensor Network Considering Obstacle,” IEEE International Conference on Advances in Computing & Communication Engineering (ICACCE-2015), Dehradun, India, pp: 249 - 252, 1-2 May 2015.

[4] A. Bakre, B.R. Badrinath, I-TCP: indirect TCP for mobile hosts, Proceedings of the 15th International Conference on Distributed Computing Systems, Vancouver, BC, May 1995, pp. 136–143.

[5] A. Cerpa, D. Estrin, ASCENT: adaptive self-configuring sensor networks topologies, UCLA Computer Science Department Technical Report UCLA/CSDTR-01-0009, May 2001.

[6] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, Habitat monitoring: application driver for wireless communications technology, ACM SIGCOMM’2000, Costa Rica, April 2001.

[7] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, A. Wang, Design considerations for distributed micro-sensor systems, Proceedings of the IEEE 1999 Custom Integrated Circuits Conference, San Diego, CA, May 1999, pp. 279–286.

[8] A. Gallais and J. Carle, “An Adaptive Localized Algorithm for Multiple Sensor Area Coverage”, IEEE 21st International Conference on Advanced Information Networking and Applications (AINA 2007) , Niagara Falls, Canada, May 2007.

[9] B.G. Celler et al., An instrumentation system for the remote monitoring of changes in functional health status of the elderly, International Conference IEEE-EMBS, New York, 1994, pp. 908–909.

[10] B. Wang, C. Fu, and H. B. Lim, “Layered Diffusion-based Coverage Control in wireless sensor networks,” Computer Networks Journal, Vol. 53, No. 7, pp. 1114–1124, 2009.

[11] Goyal, D.,Tripathy, M.R. “Routing Protocols in Wireless Sensor Networks: A Survey ,” IEEE 2nd International Conference on Advanced Computing & Communication Technologies (ACCT), Rohtak, Haryana, 7-8Jan.2012.

[12] G.D. Abowd, J.P.G. Sterbenz, Final report on the interagency workshop on research issues for smart environments,IEEE Personal Communications (October 2000) 36–40.

[13] I-F.Akyildiz, W. Su,Y.Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, Vol. 38, No. 4, pp 393–422, 2002.

[14] I.F. Akyildiz, W. Su, A power aware enhanced routing (PAER) protocol for sensor networks, Georgia Tech Technical Report, January 2002, submitted for publication.

[15] J. Agre, L. Clare, An integrated architecture for cooperative sensing networks, IEEE Computer Magazine (May 2000) 106–108.

[16] Khanouche , M.E., Ouada,F.S.,Ouguigui,S., “Energy-Aware clustering and sensor shduleding coverage maintence for wireless sensor networks,” IEEE

International Conferenece on Green Computing and Communications (GreenCom), Besancon ,20-23 Nov.2012 .

[17] Laiali Almazaydeh, Eman Abdelfattah, Manal Al- Bzoor, and Amer Al-Rahayfeh, "Performance Evaluation Of Routing Protocols In Wireless Sensor Networks," International Journal of Computer Science and Information Technology, Volume 2, Number 2, April 2010.

[18] M. Bhardwaj, T. Garnett, A.P. Chandrakasan, Upper bounds on the lifetime of sensor networks, IEEE International Conference on Communications ICC'01, Helsinki, Finland, June 2001.

[19] N. Bulusu, D. Estrin, L. Girod, J. Heidemann, Scalable coordination for wireless sensor networks: self-configuring localization systems, International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK, July 2001.

[20] P. Bauer, M. Sichitiu, R. Istepanian, K. Premaratne, The mobile patient: wireless distributed sensor networks for patient monitoring and care, Proceedings 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine, 2000, pp. 17–21.

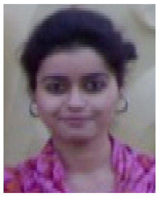
[21] P. Bonnet, J. Gehrke, P. Seshadri, Querying the physical world, IEEE Personal Communications (October 2000) 10–15.

[22] V. Tran-Quang, T. Miyoshi, "A novel gossip-based sensing coverage algorithm for dense wireless sensor networks," *Comput. Netw.* 53 (2009), pp. 2275-2287, September 2009.

[23] Vinh Iran Quang,Miyoshi T., "An Algorithm for Sensing Coverage Problem in Wireless Sensor Networks," 2008 IEEE sarnoff Symposium ,Princeton,New Jersey,USA,Paper No. S3.5,April 2008.

[24] Xin Liu, "Coverage with Connectivity in Wireless Sensor Networks",3rd international conference on broadband communications,networks and systems,2006,San jose,CA.

[25] Z. Liu, Q. Zheng, L. Xu, and X. Guan, "A distributed energy efficient clustering algorithm with improved coverage in wireless sensor networks," Future Generation Computer Systems Journal, Vol.28, No. 5, pp. 780–790, 2012.



Ms. Yogita Bahuguna obtained her B.Tech degree in AIE from DIT, Dehradun (UPTU). She received her M. TECH degree from UTU, Dehradun in Digital communication. Currently she is serving Tula's institute, Dehradun as an assistant professor in ECE department. She has an experience of more than 6.5 years in teaching and one year industry experience. She has worked for various organizations such as DIXON technologies pvt Ltd Noida, Alpine college, GRD IMT and Tula's institute, Dehradun. She has published various research papers in international conferences and journals. Her research area is Wireless Sensor Networks, Embedded System and Digital Communication.

Mr. Deepak Punetha is serving Tula's Institute, Dehradun (Uttarakhand Technical University) as an Assistant Professor in Electronics & Communication Engineering department. He has an experience of more than 5.5 years in teaching and research (Including research experience in CDAC, Mohali). He has completed his B.Tech in ECE from Dehradun Institute of Technology and M.E. (8.5 CGPA) in EPDT from PEC University of Technology, Chandigarh. He has worked for various organizations, such as Chitkara University, Himachal Pradesh (as AP), PEC University of Technology, Chandigarh (as Teaching Assistant), CDAC Mohali and Tula's Institute, Dehradun. He has published more than 35 research papers (included IEEE, Springer, Elsevier etc.) in reputed Conferences and International Journals. He is the member of reviewing committee of IEEE Explorer, Elsevier Science & Technology, Springer (Journal of Intelligent & Robotics Systems), Hindawi and various National and International Conferences and Journals. He has served as TPC member for many National & International Conferences in the region. He is also an active member of different National and International Association of Electronics and Communication Engineers and Editorial Boards. His area of interest is Electronics Product Design and Technology, Face Recognition and Compression, Radiation Pattern analysis of different Antennas, Navigation and Emergency Alerting System, Robotics and Embedded Systems.

Ms. Pooja Verma completed her B.Tech. in Electronics & Communication Engineering from N.I.E.T. Greater Noida India in 2012 and received her M.Tech degree in VLSI Design from Indira Gandhi Delhi Technical University for Women, Delhi. Currently she is serving Tula's institute, Dehradun as an assistant professor in ECE department. She has worked with C-DOT, Mehrauli and CSIR-CSIO Chandigarh as trainee. She has published various research papers in international conferences and journals. Her interest includes VERILOG FPGA Design, low power VLSI Design.

An IoT Based Predictive Connected Car Maintenance Approach

Rohit Dhall[1], Vijender Solanki[2]

[1]*Enterprise Architect, HCL Technologies, Noida, India*
[2]*Vijender Solanki, Research Scholar, Anna University, Chennai, India*

***Abstract* — Internet of Things (IoT) is fast emerging and becoming an almost basic necessity in general life. The concepts of using technology in our daily life is not new, but with the advancements in technology, the impact of technology in daily activities of a person can be seen in almost all the aspects of life. Today, all aspects of our daily life, be it health of a person, his location, movement, etc. can be monitored and analyzed using information captured from various connected devices. This paper discusses one such use case, which can be implemented by the automobile industry, using technological advancements in the areas of IoT and Analytics. 'Connected Car' is a terminology, often associated with cars and other passenger vehicles, which are capable of internet connectivity and sharing of various kinds of data with backend applications. The data being shared can be about the location and speed of the car, status of various parts/lubricants of the car, and if the car needs urgent service or not. Once data are transmitted to the backend services, various workflows can be created to take necessary actions, e.g. scheduling a service with the car service provider, or if large numbers of care are in the same location, then the traffic management system can take necessary action. 'Connected cars' can also communicate with each other, and can send alerts to each other in certain scenarios like possible crash etc. This paper talks about how the concept of 'connected cars' can be used to perform 'predictive car maintenance'. It also discusses how certain technology components, i.e., Eclipse Mosquito and Eclipse Paho can be used to implement a predictive connected car use case.**

***Keywords* — Internet of Things, Connected Cars, Predictive Maintenance, MQTT, Eclipse Mosquito, Eclipse Paho, Smart City.**

I. Introduction

The automobile and fleet management industries, majority of the consumers and the car service companies are following the 'periodic maintenance' for their automobiles. In periodic maintenance, car owners are advised to take their cars for regular service and maintenance either after certain specified time period or distance covered. For example, it is generally advised to get car serviced within three months of the last service date or after travelling 10000 kilometers, whichever comes first. Another instance where the car can be taken out for emergency service/maintenance is after some breakdown or malfunctioning of any part in the vehicle.

The way periodic maintenance works, is depicted in Fig. 1. [15]

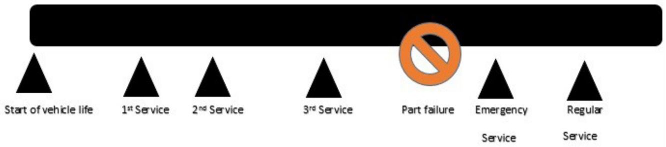


Fig. 1 Periodic Maintenance of Cars

A. Periodic Car Maintenance

Fig. 1 summarizes Periodic car maintenance. It can be explained as a service/maintenance model, where a car undergoes a service/maintenance either after a certain specified time period or on the basis of distance covered, e.g. as shown in Fig. 1, during the lifetime of a vehicle, regular services are carried out, as advised by the car manufacturer. Similarly, a car can be serviced/repaired, if there is any of the part gets faulty.

B. Drawbacks of Periodic Car Maintenance [15]

Some of the major drawbacks of periodic car maintenance are listed below:

- Higher cost of service, as vehicles are required to be get serviced as per the schedule
- Even if vehicle/parts are in perfect health, still service needs to done and parts to be replaced
- No way of knowing, if a part needs immediate attention, and can result in breakdown of the vehicle
- This breakdown could cost significant charges for the car owners

II. Alternate Approach to Periodic Car Maintenance

Instead of getting a car serviced periodically, if a system developed using sensors and IoT [9] technology stack is used, which collect and analyze fitness and running condition of different parts of the car, and send this data to a centralized system. In this centralized system, data received from these connected cars, can be analyzed further and if any service is needed, a service request can be raised. This proposed system can also generate emergency alerts, in case any part is about to break down, thus avoiding car/part failure [15][17]

A. Advantage of Proposed system [18]

- Reduction in service and maintenance costs, as only parts which needs to be replaced or serviced
- Real time alerts of possible part failure, thus avoiding breakdown and costs associated with outages
- Analytics and reporting dashboards can be used to view how the car is performing over different periods of time and in different locations
- Driver's driving habits can be analyzed and appropriate action can be taken
- Tour and cab providers can manage their fleet better, thus maximizing profits
- Target advertisements for monetization of data received from connected cars (e.g. offering service discounts for car which needs to be serviced etc.)

DOI: 10.9781/ijimai.2017.433

III. What is Predictive Car Maintenance and why we need it?

A common practice, generally followed in automobile world is 'periodic car maintenance'. In this, the car is supposed to undergo periodic service and maintenance routine. When to get a car serviced, is generally decided by either a specified time period or distance covered. For example, it is generally advised to get the car serviced within six months of the last service date or after travelling 10000 kilometers, whichever happens first.

Now, the problem with 'periodic car maintenance' is that nobody is sure if any part or lubricants really needs to be serviced/replaced. This normally leads to parts/lubricants, which are in good condition, getting changed/serviced. Another problem, which is generally faced is that though scheduled 'periodic car maintenance' is still some time away, there is some problem with a part, which needs immediate attention. 'Periodic car maintenance' cannot solve this problem, and only way to know about this is after break down. So, two problems, associated with this model of car servicing can be summarized as:

- Higher service costs, as parts which are fine, will also be replaced
- Unable to generate any alert, if any part needs immediate attention/ service, resulting in breakdown/outages

This is where 'Connected Cars' and 'Predictive car Maintenance' can help [15][16]. Connected cars can collect data, from different sensors installed in the car, related to the health status of different parts, and send it over the internet to backend applications, for analytical and decision making purposes. One of the backend analytical application, based on the health status of different parts, can invoke a workflow and schedule an appointment with the service provider, if some part needs immediate attention. Similarly, real time alerts can be sent to concerned parties, in case something need immediate attention.

This can result in considerable savings in terms of service and maintenance charges of the car [18]. Now, only the parts which actually need replacement, will be serviced. This data will be collected and transmitted by different sensors fitted on the car for performing health check of different parts oil health check, tire and pressure health check, filters health check and so on.

IV. How this Work Differs from Other Work Done in this Area

Good amount of research work is done on the Predictive maintenance topic [1][4][6]. Some of the work talk about how to collect or read sensor data (from cars, from manufacturing industrial machines etc.), or propose a model to perform predictive analysis and so on [2][3] [5][7]. This work proposes an IoT based approach [15] to collect this data, send it to the cloud and perform predictive analytics on this huge amount of data. The proposed approach is based on industry proven protocols and products, some of which are Eclipse IoT's [11] Eclipse Mosquitto [13], Eclipse Paho[12] and MQTT[10] protocol. High level IT architecture is also provided, so that same can be referenced by people to build, extend and further improve the system based on this architecture.

Finally, a simulation of the proposed architecture model is also given, where a client GUI utility simulates the car sensor, and sends data to the cloud.

V. Proposed Technology for Implementing Predictive Connected Car Maintenance

This section introduces some of the important technology components, which will be used in the proposed implementation of 'Connected car' use case for predictive maintenance.

A. MQTT Protocol

Message Queue Telemetry Transport (MQTT) is a light-weight messaging protocol based on publish-subscribe model. MQTT uses a client-server architecture where the client (such as a sensor device on cars) connects to the MQTT server (called a broker) and publishes messages to server topics. The broker forwards the messages to the clients subscribed to topics. MQTT is well suited for constrained environments where the devices have limited processing and memory resources and the network bandwidth is low.

B. Deeper look into MQTT

MQTT is an extremely lightweight messaging protocol. Its publish/ subscribe architecture is designed to be open and easy to implement. Single MQTT server can support up to thousands of remote clients. These characteristics make MQTT ideal for use in constrained environments where network bandwidth is low or remote devices that might have limited processing capabilities and memory, need to be supported. The MQTT protocol is based on publish/subscribe model. Publishers can send the messages to the topics, configured on the MQTT server (also called MQTT broker). Clients can subscribe to these topics and receive whatever messages are published on those topics.

Fig. 2 depicts the publish/subscribe model of MQTT

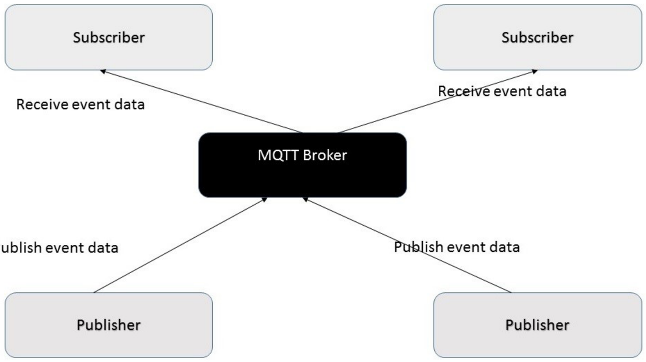


Fig. 2 Publish/Subscribe model of MQTT.

Though MQTT's publish-subscribe model is identical to any existing enterprise messaging systems, the main advantage of MQTT has over fully blown "enterprise messaging" systems are that its low footprint makes it ideal for developing IoT applications with small sensors, devices and other low-capacity things. For example, Facebook uses MQTT for its messenger product on the mobile platform, to ensure that battery usage of this application is minimized.

Some of the major advantages of MQTT are listed below:

- Publish Subscribe model provides one-to-many message delivery
- Uses TCP/IP for network connectivity
- Can work with SSL/TLS for security
- MQTT offers three message delivery QoS: 1) at most once ,2) at least once and 3) exactly once
- These QoS are met even in case of network, publisher or client failures
- Very simple specification and APIs, making it easier for developers to work with MQTT based products
- Most important APIs are CONNECT, PUBLISH, SUBSCRIBE, UNSUBSCRIBE, and DISCONNECT
- As MQTT is specifically designed for constrained device, it provides only the bare minimum features to support them.
- The message header is short in MQTT and smallest packet size in 2 bytes, making it ideal for small and constrained devices

- As MQTT is a publisher/subscribe model, sender and receivers are decoupled from each other
- Doesn't restrict the format of data to be in any particular format, thus allowing flexibility
- 'Last Will' feature, which allows abnormal client/sensor termination to be notified to all interested parties
- Both commercial and open sources MQTT based broker products are available. These include IBM WebSphere MQ v 7.1 onwards, EclipseIoT Mosquitto, ActiveMQ and HiveMQ.

C. Eclipse Mosquitto

Eclipse Mosquitto is an open source MQTT broker/server. Based on the lightweight MQTT protocol, Mosquitto is ideal for devices, sensors and other 'Internet of Things' devices, with low processing capacity. MQTT clients can connect to a given Mosquitto broker and publish/subscribe the messages from a topic.

Eclipse Mosquitto's main responsibility is to provide a communication channel between publishers/senders and subscribers/receivers. Any publisher, using the Eclipse Paho MQTT Client API can publish the messages to an MQTT Broker. These MQTT clients should specify the topic, on which they want to publish the message. These topics are configured on MQTT broker. Any subscriber or receiver, that want to receive the message, subscribe to that particular topic. It is the responsibility of the broker to deliver all the messages arriving on a topic to all interested clients. As different clients (both publishers as well as subscribers) need to know only broker/topic details, both are decoupled from each other. This architecture pattern has many advantages, e.g. highly scalable solution, where subscribers needn't to be overwhelmed by publishers sending messages at a rate faster than what a subscriber can process.

D. Eclipse Paho

Eclipse Paho is an EclipseIoT project and is implementation of MQTT protocols. Eclipse Paho provides MQTT client libraries in multiple languages including Java/C++, C#, .NET and Python. Eclipse Paho also has utilities for MQTT-SN (sensor networks). Both publishers and subscribers (as shown in Fig. 2) can use API's provided by Eclipse Paho MQTT Client library, and send/receive messages to/from MQTT broker (e.g. Eclipse Mosquitto).

VI. Why MQTT and Other Proposed Technology Components in Connected Car Implementation

- Suitable for low capacity devices like sensors fitted on connected cars
- Provides Quality of services to handle connectivity and other errors, which can be quite common in the case of cars and automobiles, which are on the move, and n/w connectivity can be an issue
- Supports wide variety of languages, so compatibility will not be an issue for any existing technology platform of a car manufacturer
- Also integrated with proven and well adopted industry leading messaging systems like WebSphere MQ and ActiveMQ
- Message formats can be customized, allowing manufacturer to customize and innovate the solutions

Details of MQTT protocol and its specification can be found on the MQTT site, given in the reference section.

VII. Proposed Architecture of 'Predictive Car Maintenance' using Eclipse Mosquitto and Eclipse Paho

Fig. 3 shows the simplistic high level architecture context diagram of a system implementation of 'predictive car maintenance' using Eclipse Mosquitto and Eclipse Paho.

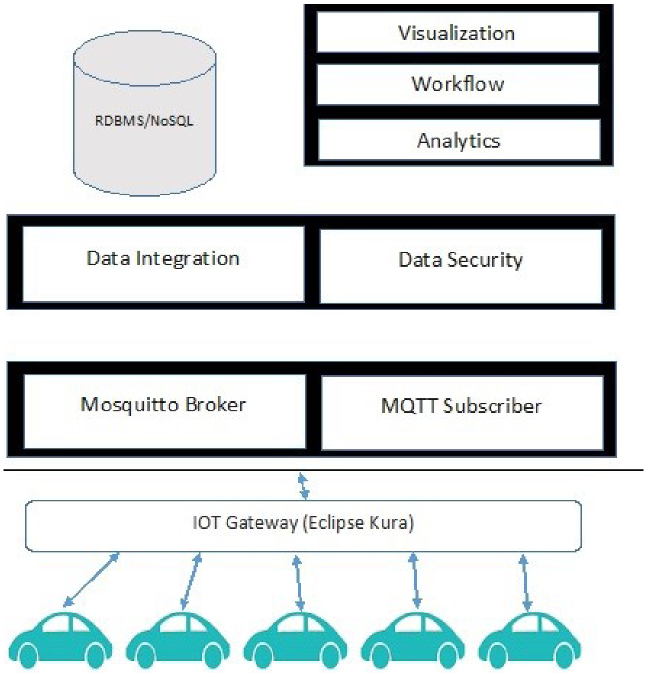


Fig. 3 Architecture diagram of 'Connected car' ecosystem.

Flow of context diagram (Fig. 3) can be explained as follow:

- 'Connected Cars' send data in predefined format to IoT Gateways like Eclipse Kura.
- Cars can use any possible way to send the data, i.e. via Wi-Fi, Telco services etc.
- IoT Gateway would send this data to MQTT based Eclipse Mosquitto Broker hosted in a cloud environment
- In many scenarios, there can be additional components like coordinator/controller nodes in the architecture, which do some kind of pre-processing/aggregation of data collected from various devices, before sending it to the cloud
- Once data is received by Eclipse Mosquitto, subscribers will receive this message, using Eclipse Paho API
- After doing basic validations and any data conversion, subscribers can send this message to downstream systems, using services exposed by the Data Integration component of the architecture
- It could be that these messages are sent to a messaging system, e.g. Apache Kafka, where these messages can be consumed by the workers
- These workers can push these messages to Apache Hadoop or other such data processing service, for analytical purpose
- These messages can also be consumed by data processing services, handling real time stream of messages e.g. Apache Storm
- These real time stream of messages can be used for real time analytics purpose e.g. sending alerts in case something needs immediate attention

'Workflow' component can be used to define and execute different workflows, based on some conditions

For example, to schedule a service appointment, invoke a REST based service of the car service provider, in case some parts need to be replaced.

There can be various visualization tools to view reports of summarized data and perform analytical queries.

Note that, in real life complex scenarios, there can be many more components involved in the architecture (e.g. configuration services, policy manager, rule engine and so on), but for simplicity's sake, those have not been included in this paper.

Fig. 4 summarize the high level data flow of the proposed system.

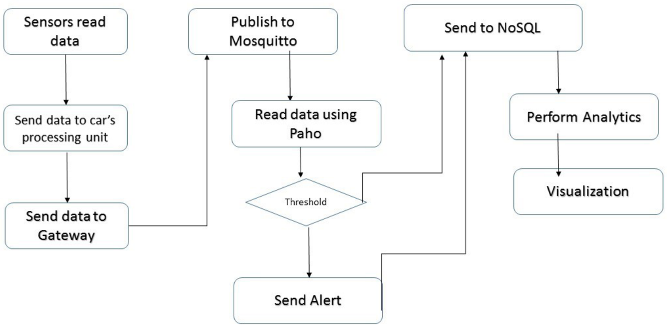


Fig. 4 Sample data flow for connected car.

VIII. Sample Implementation of Predictive Car Maintenance Use case with Eclipse Mosquitto and Eclipse Paho Client Utility

In this section will use Eclipse Paho Client utility to simulate a connected car, which will send city where the car is (can send exact location also), speed and current car health check, including if any part needs to be replaced or not.

In the real world, devices like connected cars might be sending data first to a IoT gateway, as shown in architecture diagram in Fig. 3 in last section, where this data will be processed and aggregated, before being fed to further downstream applications. Depending upon the actual requirement, applications can be designed to take appropriate actions based on the data being received e.g. Send alerts if speed is too high or schedule an appointment with the car service agency, if some part/lubricants needs to be replaced.

Sample format of the MQTT message, transmitted by sensor fitted on the car is shown on Fig. 5.

```
CarId=xxxxxx;Location=XXXX;ItemName=<<Value>>;currentS
tatus=<<value>>
```

Fig. 5 Sample MQTT message format, sent by connected car

Some of the information, which can be sent by the connected car is shown in Table 1.

TABLE I.
LIST OF SAMPLE PARAMETERS WHICH ARE MONITORED

Parameters
Location
Mileage per litre
Quantity of fuel consumed
Total Distance Covered
Trip Distance
Distance covered in top gear
Top Speed
Fuel Level
Current temperature
Coolant Level
Engine Oil level
Fuel Level

In this paper, we will be simulating the behavior of connected car, using Eclipse Paho MQTT Utility, a Java Swing based GUI application, to connect to a Mosquitto server, publishing message to a topic on Mosquitto server. The client will receive this message and for simplicity, will display this message content in the GUI.

Once the utility is launched, screen as shown in Fig. 6 will appear. Specify the address of the MQTT Mosquitto server and port and connect to the server. We are using a cloud based test Mosquitto server, available for public, at the address "test.mosquitto.org", port 1883.

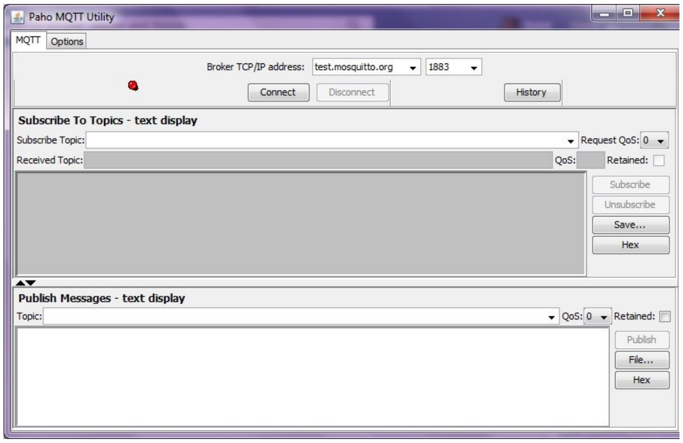

Broker TCP/IP address: test.mosquitto.org 1883
Request QoS: 0
QoS: 0

Fig. 6 Connecting to MQTT server using Paho MQTT utility.

After specifying the address, click "Connect". If everything goes right, you will be connected to server, else you will get error message.

Once connected, enter the name of the topic, and click subscribe. Now, any message, published on this topic will be displayed in the text area of the subscriber. Fig. 7 shows a subscriber, connected to a topic named 'CarHealth'.

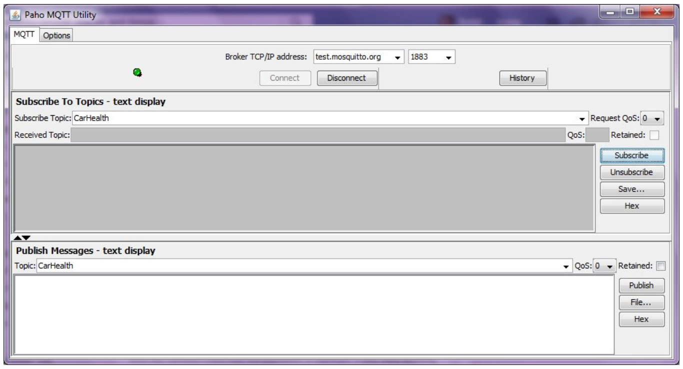
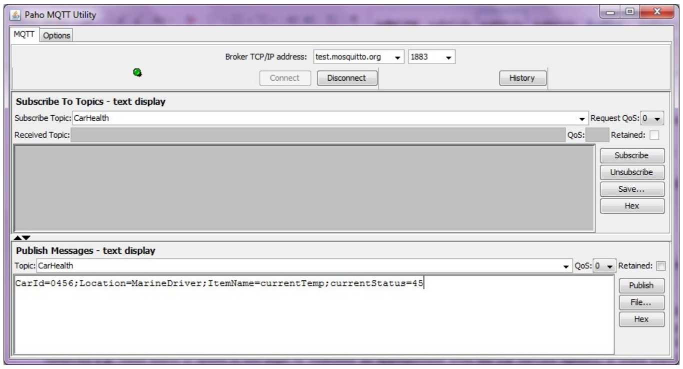

Broker TCP/IP address: test.mosquitto.org 1883
Subscribe Topic: CarHealth
Request QoS: 0
Topic: CarHealth
QoS: 0

Fig. 7 Subscribing to a MQTT topic.

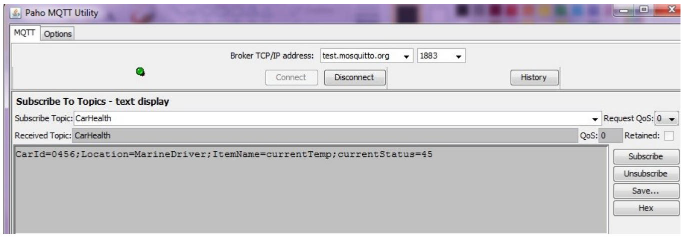

Broker TCP/IP address: test.mosquitto.org 1883
Subscribe Topic: CarHealth
Request QoS: 0
Topic: CarHealth
QoS: 0

```
CarId=0456;Location=MarineDriver;ItemName=currentTemp;currentStatus=45
```

Fig. 8 Publishing a message to MQTT topic.

Enter the message payload in the 'Publish Messages – text display' area and click publish. Fig. 8 shows the step to publish a message onto MQTT topic.

Once the message is published, all clients who have subscribed to this topic, will receive this message. In our scenario, this will be displayed in the utility GUI. Fig. 9 shows the scenario of a subscriber receiving the message.



Paho MQTT Utility

Broker TCP/IP address: test.mosquitto.org 1883

Subscribe To Topics - text display

Subscribe Topic: CarHealth Request QoS: 0

Received Topic: CarHealth QoS: 0 Retained: □

```
CarId=0456;Location=MarineDriver;ItemName=currentTemp;currentStatus=45
```

Fig. 9 Subscriber receiving the message

IX. Cost benefit of Predictive Car Maintenance

Predictive car maintenance can help address the issues of traditional periodic car maintenance approach. Some of the advantages it provides are

- Reduction in service and maintenance costs, as only parts which needs to be replaced are serviced
- Real time alerts of possible part failure, thus avoiding breakdown and costs associated with outages
- Analytics and reporting dashboards can be used to view how the car is performing over different periods of time and in different locations
- Driver's driving habits can be analyzed and appropriate action can be taken
- Tour and cab providers can manage their fleet better, thus maximizing profits
- Target advertisements for monetization of data received from connected cars (e.g. offering service discounts for car which needs to be serviced etc.)

Table 2 shows the cost comparisons of periodic vs predictive maintenance for a medium sedan car. Periodic service cost figures are taken from a leading automobile web site (see reference). As most of the car vendors provide first two services as free services, zero cost is taken for these two services. For predictive maintenance, it is assumed that service cost will go down by 30%.

TABLE II. Service Cost Comparison of Periodic vs Predictive Car Maintenance

Service	Cost (Periodic Maintenance) in INR	Cost (Predictive Maintenance) in INR
1st Service	0	0
2nd service	0	0
3rd service	2465	1726
4th service	6455	4519
5th service	3835	2685
6th service	6455	4519
7th service	3835	2685
8th service	6455	4519
Total	29500	20653

Sample calculation for 3rd service is as follow

Cost of 3rd periodic service – 2465 INR [21]

Cost with predictive maintenance with 30% saving = 2465 *((100-30)/100) = 1726 INR

Fig. 10 represents another view of the service cost comparison data of Table 2. During the first eight services (scattered across 5 years) of the car, total costs incurred on periodic maintenance is 29500 INR. This cost will come down to 20653 INR (assuming 30% reduction in service cost), with the help of predictive car maintenance.

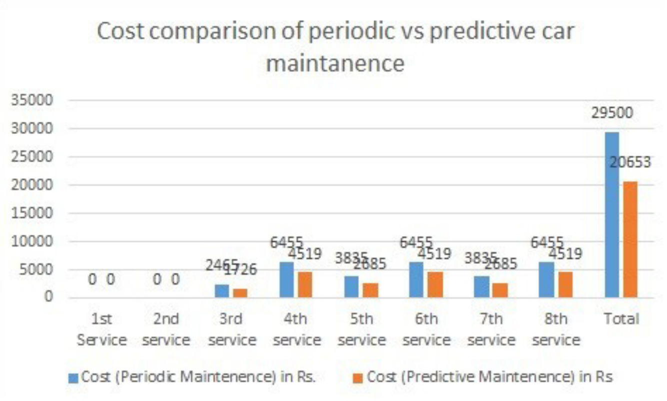


Fig. 10 Service cost comparison of periodic vs predictive car maintenance.

To understand the outage costs of a fleet/transport organization, let us take an example of a company with a fleet strength of 100 cars. If a car of such commercial organization is out of service because of breakdown or faulty part, there will be various costs associated with it e.g. pay for an unutilized driver and support staff, rental of the car for that day, fixing and service cost, need to ensure alternate vehicle for the customer to ensure company's commitment, else loses on reputation part in the consumer market and so on.

Assume that the total outage cost for one vehicle going out of service is Rs. 5000 per day. So, for a company with one vehicle out of service, associated costs can be calculated as follows

Outage cost of one vehicle going out of service – 5000 INR (A)

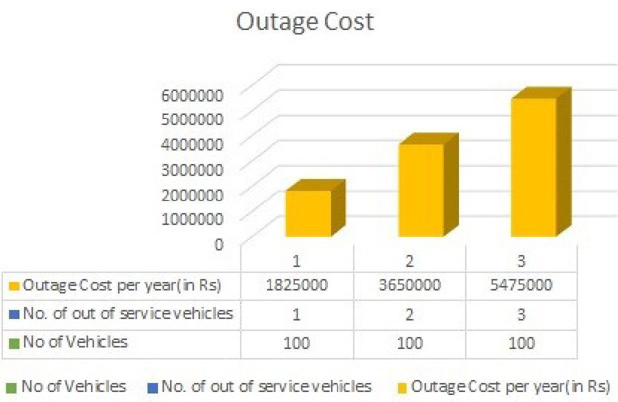
Annual cost of one vehicle going out of service – 5000(A) * 365 = 1825000 INR (B)

Table 3 shows the outage cost per year for multiple number of vehicles going out of service on a given day. For a company with a fleet size of 100, number of vehicles going out of service can be much higher, but for simplicity, table 3 shows costs for maximum three vehicles going out of service.

TABLE III. Outage/Breakdown Cost for a Fleet/Transport Company

No of Vehicles	No. of out of service vehicles	Outage Cost per year(in INR)
100	1	1825000
100	2	3650000
100	3	5475000

Fig. 11 represents another view of the outage/breakdown cost data of Table 3. Assuming that per day cost of a vehicle going out of service is 5000 INR. As shown, on average with only one vehicle out of service, cost per year is 1.8 million INR and with three vehicles going out of service, this will increase to 5.4 million INR per annum.



	1	2	3
🟨 Outage Cost per year(in Rs)	1825000	3650000	5475000
🟦 No. of out of service vehicles	1	2	3
🟩 No of Vehicles	100	100	100

Fig. 11 Outage/Breakdown cost for a fleet/transport company.

Table 4 shows the outage cost comparison with 30% improvement in outage scenarios.

Original Outage cost of 1 vehicle out of service = 1825000 Rs (B)

Cost after 30% improvement in outage situations = 1825000 * ((100-30)/100) = 1277500 Rs.

TABLE IV BREAKDOWN/OUTAGE COST COMPARISON

No of Vehicles	No. of out of service vehicles	Outage Cost per year(in INR)	Outage cost @30% improvement
100	1	1825000	1277500
100	2	3650000	2555000
100	3	5475000	3832500

Fig. 12 represents another view of the outage/breakdown cost comparison data of Table 4. Even 30% assumed reduction in downtime will provide considerable savings for the organization. For three vehicles out of service, cost of breakdown will come down from 5.4 million INR to 3.8 million INR.

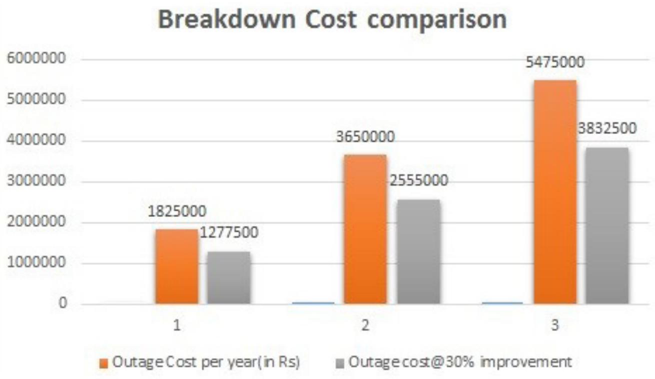
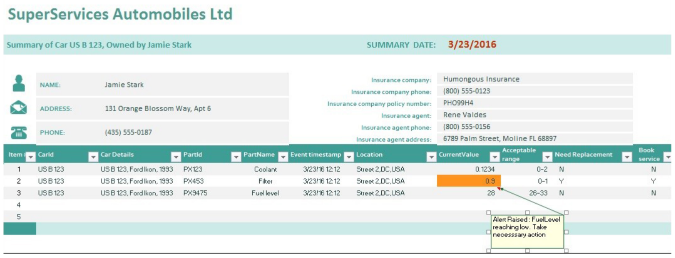


Fig. 12 Breakdown/outage cost comparison.

Fig. 13 depicts a sample portal/dashboard to view the status of a given car. Any authorized person can view, whether any part needs replacement or not, current status of different parts, historical information, including details of any alert that was sent.

For example, the second row in this dashboard shows that an alert was raised for a particular part. Though the current value of this part is within a valid range (between 0-1), but as it is almost on the threshold to breach the value, a pro-active alert was sent, thus avoiding any possible outage and associated costs



SuperServices Automobiles Ltd

Summary of Car US B 123, Owned by Jamie Stark SUMMARY DATE: 3/23/2016

NAME: Jamie Stark
ADDRESS: 131 Orange Blossom Way, Apt 6
PHONE: (435) 555-0187

Insurance company: Humongous Insurance
Insurance company phone: (800) 555-0123
Insurance company policy number: PHO99H4
Insurance agent: Rene Valdes
Insurance agent phone: (800) 555-0156
Insurance agent address: 6789 Palm Street, Moline FL 68897

Item #	CarId	Car Details	PartId	PartName	Event timestamp	Location	CurrentValue	Acceptable range	Need Replacement	Book service
1	US B 123	US B 123, Ford Ikon, 1993	PX123	Coolant	3/23/16 12:12	Street 2,DC,USA	0.1234	0-2	N	N
2	US B 123	US B 123, Ford Ikon, 1993	PX453	Filter	3/23/16 12:12	Street 2,DC,USA	0.9	0-1	Y	Y
3	US B 123	US B 123, Ford Ikon, 1993	PX9475	Fuel level	3/23/16 12:12	Street 2,DC,USA	28	26-33	N	N
4										
5										

Alert Raised : FuelLevel reaching low. Take necesssary action

Fig. 13 Sample dashboard for viewing parts/items health.

This dashboard can also be used to trigger additional workflows. For example, a request to book a service appointment with the car service provider can be raised using this portal.

X. Challenges in the Predictive Maintenance of Connected Cars

Some of the challenges in successful implementation of predictive maintenance and connected cars are as follows [8]:

- Govt. and Regulatory policies, restricting on what kind of sensitive data can be transferred
- Security concerns related to location and other sensitive data being shared and transmitted
- Lack of industry standards. Right now, most of the work done is vendor specific/proprietary
- Need to have a proper IT Analytics system in place. Can involve huge costs upfront
- Need better connectivity in terms of telecom, Bluetooth, Wi-Fi and other such networks for transmission of real time data from sensors
- Associated business use cases are still evolving, so justifying initial costs can be difficult
- With higher number of sensors needed on the car, cost of buying a new car can go up

XI. Conclusion

‘Connected car’ concept is getting lots of traction with automobile companies these days. There are multiple benefits of ‘Connected Car’ ecosystem, and one such benefit is Predictive Car Maintenance. This paper talked about what predictive car maintenance is all about, which problems it could solve. MQTT, a popular protocol for IoT is also discussed, followed by an introduction to Eclipse Mosquitto and Eclipse Paho, an implementation of MQTT.

This paper also presented a high level architecture of how Connected car use can be implemented, using Eclipse Paho and Eclipse Mosquitto. A simulation of ‘connected car’ sending sensor’s data to the cloud is also discussed. Finally, cost saving of a predictive car maintenance system over a traditional periodic car maintenance system is shown. This paper concluded by sharing of some of the challenges in implementing predictive maintenance of connected cars.

References

[1] Kevin A. Kaiser ; Cerner Corp., Kansas City, MO ; Nagi Z. Gebraeel ,Predictive Maintenance Management Using Sensor-Based Degradation Models, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans (Volume:39,Issue: 4),pp 840-849 ,2009

[2] D. C. Swanson ; Appl. Res. Lab., Pennsylvania State Univ., University Park, PA, USA ,A general prognostic tracking algorithm for predictive maintenance , Aerospace Conference, 2001, IEEE Proceedings. (Volume:6) pp 2971 - 2977 vol.6 ,(2001)

[3] A. Grall ; Lab. de Modelization et Surete des Systemes, Univ. de

Technologie de Troyes, France ,L. Dieulle , C. Berenguer and M. Roussignol ,Continuous-time predictive-maintenance scheduling for a deteriorating system,IEEE Transactions on Reliability(Volume:51 ,Issue: 2),pp 141 - 150 , 2002

[4] Stabler, L,Hawthorne, K(2004) FIX IT BEFORE IT'S BROKE, Railway Age , Volume: 205, Issue Number: 9 ,pp. 83-84,2004-9

[5] J. Endrenyi , Ontario Power Technol., S. Aboresheid , R. N. Allan , G. J. Anders , The present status of maintenance strategies and the impact of maintenance on reliability,IEEE Transactions on Power Systems(Volume:16 ,Issue: 4),pp 638 - 646 ,2001

[6] Joel Levitt,Complete Guide to Predictive and Preventive Maintenance ,Industrial Press, Inc.; 2 edition (June 15, 2011)

[7] Joseph S. Ng,Automated wireless preventive maintenance monitoring system for magnetic levitation (MAGLEV) trains and other vehicles http://www.google.com/patents/US5445347, 1995

[8] Why Do Predictive Maintenance Programs Fail? - http://reliabilityweb.com/articles/entry/why_do_predictive_maintenance_programs_fail/

[9] What is Internet of things - http://whatis.techtarget.com/definition/Internet-of-Things

[10] Details about MQTT protocol - http://mqtt.org/

[11] Eclipse IoT - http://iot.eclipse.org/

[12] Eclipse Paho - http://www.eclipse.org/paho/

[13] Eclipse Mosquitto - http://projects.eclipse.org/projects/technology.mosquitto

[14] Eclipse Paho MQTT Client API – http://www.eclipse.org/paho/files/javadoc/index.html

[15] IoT, Analytics & Cars – Joe Speed - https://mobilebit.wordpress.com/

[16] Practical MQTT with Paho- http://www.infoq.com/articles/practical-mqtt-with-paho

[17] IoT and Predictive Maintenance- http://blog.bosch-si.com/categories/manufacturing/2013/02/iot-and-predictive-maintenance/

[18] The Smart and Connected Vehicle and the Internet of Things - http://tf.nist.gov/seminars/WSTS/PDFs/1-0_Cisco_FBonomi_ConnectedVehicles.pdf

[19] IBM Predictive Maintenance and Quality for automotive - http://www.novemba.de/wp-content/uploads/IBM-Predictive-maintenance-and-Auality-for-automotive.pdf

[20] The advantages of implementing a predictive management within the maintaining and equipment repair at an enterprise which produces components for the automotive including technical assistance and services - http://eccsf.ulbsibiu.ro/repec/blg/journl/5317sima.pdfconnected cars - use cases for Indian scenario - http://www.hcltech.com/white-papers/engineering-and-rd-services/connected-cars-use-cases-indian-scenario

[21] Maruti Swift Diesel Estimated Maintenance Cost - https://www.cardekho.com/maruti-swift/service-cost.htm

Rohit Dhall is working as an Enterprise Architect with Engineering and R & D Services,HCL Technologies,India. He has over 19 years of software industry experience. He helps global clients build technical solutions to solve their complex business problems. His main area of expertise is architecting, designing and implementing high performance, fault tolerant and highly available solutions for leading Telco and BFSI organizations. He has worked on diverse technologies like java/J2ee, client-server,P2P ,DWH,SOA, BigData and IoT etc. He regularly writes articles, blogs and white papers for various IT forums, portals and events. He is also a coauthor of IBM Redbook and Redpaper on 'ITCAM for WebSphere'.

Vijender Kr. Solanki, Ph.D is a research scholar in the department of Computer Science and Engineering at Anna University, Chennai. He has completed his graduation and postgraduation (B.Sc., M.C.A and M.E) from institution affiliated with Maharishi Dayanand University, Rohtak (MDU) Haryana, India in 2001, 2004 and 2007 respectively. He has attended an orientation program at UGC-Academic Staff College, University of Kerala and a refresher course at IIIT-Allahabad. He has participated in more than 15 seminars, summits and conferences at various national & international levels, including IIT-Delhi, Bharathiar University, Coimbatore and Anna University, Chennai. He has published more than 10 technical papers with IEEE, Springer and Elsevier- Science-Direct library. His research interest includes smart city, network security and network management. He is having 08 Years of rich academic experience. He has delivered many technical lectures in various institutions including AICTE Sponsored SDP-FDP Lectures at SKNCOE, Pune, SNS College, Coimbatore, ITS Ghaziabad and DAVIM, Faridabad. He was an invitee as key note speaker in DST Sponsored seminar at RCEW, Jaipur. He has chaired session in many conferences. He is reviewer of some of the IEEE, Springer and Elsevier Journals and Conferences which are indexed in Scopus, DBLP, ACM Digital library. He is also a book editor with Universities Press.

Performance Evaluation of AODV Routing Protocol in VANET with NS2

Ms. Divya Rathi[1] Mrs. R.R.Welekar[2]

[1]SRCOEM, CSE Department, M.Tech Scholar, Nagpur, India
[2]SRCOEM, CSE Department, Assistant Professor, Nagpur, India

***Abstract* — In intelligent transportation systems, the collaboration between vehicles and the road side units is essential to bring these systems to realization. The emerging Vehicular Ad Hoc Network (VANET) is becoming more and more important as it provides intelligent transportation application, comfort, safety, entertainment for people in vehicles. In order to provide stable routes and to get good performance in VANET, there is a need of proper routing protocols must be designed. In this paper, we are working with the very well-known ad-hoc on-demand distance vector (AODV) routing protocol. The existing Routing protocol AODV-L which is based on the Link expiration time is extended to propose a more reliable AODV-AD which is based on multichannel MAC protocol. For the performance evaluation of routing protocols, a simulation tool 'NS2' has been used. Simulation results show that the proposed AODV-AD protocol can achieves better performances in forms of high Route stability, Packet Delivery ratio and packet loss rate than traditional AODV-L and traditional AODV.**

***Keywords* — VANET, AODV, NS2, Packet Delivery Ratio, Packet Loss Ratio, Route Reliability.**

I. Introduction

There are various types of ad hoc networks and one of them is VANET. Vehicular Ad Hoc Network (VANET) is a subclass of Mobile Ad hoc Network (MANET). MANET and VANET have some common features such as low bandwidth and self-organization and shared radio transmission. The main work of VANET is the provision of vehicle-vehicle wireless communication and vehicle infrastructure communication (e.g., between vehicles and road side equipment), and these connections can be established without central access. The communication between vehicles has some specifics such as high speed and mobility, and that is the key feature of vehicular ad-hoc networks that makes them unique in the context of MANETs. By using vehicle to vehicle communications, drivers can be notified of important traffic data such as the condition of roads and accidents. Such information will improve drivers' decisions in hard conditions. Moreover, vehicular communications will help to monitor and manage traffic distribution and to improve vehicle fuel economy. Routing algorithms are an important part of a vehicular ad hoc network where it forward information in order to connect vehicles and having communication between them. The proposed routing protocols and mechanisms that may be employed in VANETs should adapt to the rapidly changing topology.

In this paper, we propose a new routing scheme to make a more reliable route from source to destination node. The whole work is based on the VANET scenario, where vehicles move with different velocities along two directions on the highway. The simulation is performed to evaluate the performance of the proposed algorithm in comparison to existing AODV-L and the traditional AODV routing protocol. Packet delivery ratio and Loss packet ratio are the performance metrics considered in the evaluation process.

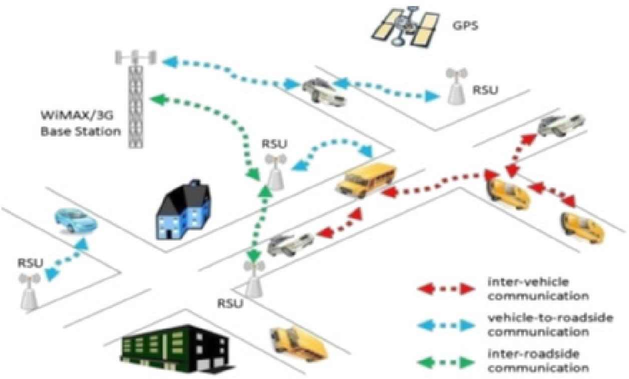

GPS
RSU
WiMAX/3G Base Station
RSU
RSU
RSU
inter-vehicle communication
vehicle-to-roadside communication
inter-roadside communication

Fig1. Architecture of Vehicular Network.

A. *AODV routing protocol:*

AODV routing protocol is classified as a member of Bellman-Ford distant vector protocol which work in mobile network. AODV is a reactive and an on demand distance vector routing protocol, which means that that it searches a route only when there is a need of sending data packets from source to a destination. It distributes routing request packets whenever it is required due to which network overhead is very low along with that it provides loop-free routes. This protocol uses following Messages for transmission:

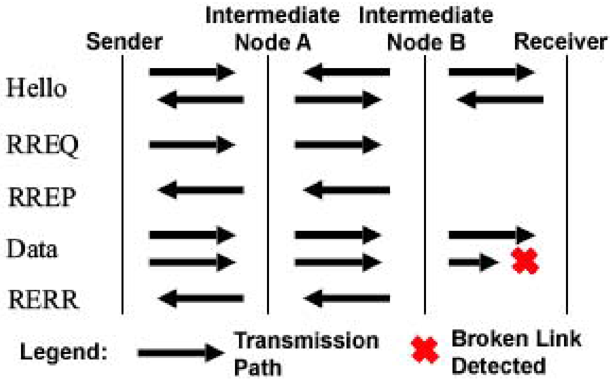


Fig 2. AODV routing message

A RREQ message is broadcasted when a source node needs to discover a route to a destination. As a RREQ propagates the intermediate nodes of the network updates their routing tables by using

DOI: 10.9781/ijimai.2017.434

it. The RREQ also contains the most recent sequence number for the destination. When a RREQ reaches to the destination node, a route is made available by unicasting a RREP message back to the source. If it is itself the destination, then a node generates a RREP. As the RREP propagates back to the source node, middle nodes update their routing tables in the direction of the destination node direction. RERR message is broadcast for broken links. It is generated directly by a node or passed on when received from another node.

II. Literature Survey

In [1], author Yang He, Wenjun Xu and Xuehong Lin, proposes a new stable routing protocol, which is based on the scenario where vehicles move at a different velocity on the highway. The uniqueness of this work lies in its specific design that considers the vehicular motion and the channel state information between all vehicles which wants to establish links. In this way, the communication-link reliability is improved by calculating the link expiration time among the vehicles of the route from source to destination. The simulation experiments have been performed to calculate the performance of the proposed scheme in comparison to the AODV protocol. Finally, the improvement of the AODV-L is evaluated in the terms of the performance metrics packet delivery ratio and average end-to-end delay.

The survey paper [2] provides a broad overview in vehicular networking and gives a brief introduction to the limitations of the routing protocols and the challenges in designing algorithms for VANETs is to provide reliable packet transmission with minimum delay, maximum throughput, and low communication in vehicles. The authors of [3] propose a reliability-based routing system making an allowance for the mathematical distributions of movement of vehicles and the link breakages in the route. In [4], authors propose a routing protocol by make use of the vehicles movement information (e.g., position, velocity, speed, acceleration and direction) based on Ad-hoc On-Demand Distance Vector (AODV) [5] and gives its significance. In [6], the author takes on the stochastic large-scale fading model which gives results in a log-normal formula as a naive channel model.

In Hop Reservation Multiple Access [7] describes a multi-channel protocol which uses slow frequency hopping spread spectrum (FHSS) for the hosts hop from one channel to another channel in network as per to a predefined hopping configuration. First the two nodes do the handshaking by RTS/CTS and exchange the data by it, for the communication they stay in a frequency hop. Other hosts carry on hopping, and on different frequency hops more than one communication can be take place. In [8] Receiver Initiated Channel-Hopping with Dual Polling uses a same approach, but the receiver node initiates for the avoidance of the collision and do handshake process in the place of the sender. This can be done by using only one transceiver at each host in the network, but only frequency hopping networks can be applied and cannot be used in systems which use the mechanisms like direct sequence spread spectrum (DSSS). Nasipuri et al. [9] propose a multi-channel CSMA protocol with "soft" channel reservation. If there are N channels, then according to protocol each host can listen concurrently to N channels. If there is an idle channel, then a host which wants to transmit packet select that channel. The preference is given to that channel which was used for the last successful transmission. In [10] the protocol is extended in order to select the preeminent channel based on signal power detected at the sender. These protocols require N number of transceivers for each host, which is very expensive this is drawback to this protocol.

III. Research methodology

In this paper we are introducing a new system which is the enhanced part of the existing system which is based on the concept of Link Expiration Time (LET). As described further the existing system work is done in OMNET++ and we are doing work with NS2 simulator. So, whole work is divided in two parts:

1. Firstly, there is a comparison between the traditional AODV and the AODV-L routing algorithm using NS2 simulator.
2. Then making of enhanced algorithm which is the extended part of the AODV-L algorithm with the concept of multichannel MAC protocol.

A. *Existing System:*

In this system by using the vehicles movement information with highway mobility model based on position, direction, velocity it is predicted that how long the route is reliable [1]. There is a use stochastic large-scale fading channel model to calculate the transmission range, which should also be stochastic and the highway mobility model is used to recalculate the route lifetime. In this way, a new reliability model is proposed to facilitate the reliable routing in VANETs. The well-known ad-hoc on-demand distance vector (AODV) routing protocol is extended to propose the reliable ad-hoc on-demand distance vector routing protocol AODV-L. Simulation results is demonstrated by OMNET++ that AODVL outdoes significantly the AODV routing protocol in terms of more efficient delivery ratio and less end-to-end delay.

1) *Channel Model:*

Stochastic Large-Scale Fading Model:

Roadside constructions, foot-travelers and vehicles themselves may become difficulties in communication in VANET which affects the channel state among the vehicles. Moreover, due to the continuous moment of the vehicles the transmission environment also varies. Considering all these factors, there is a stochastic large scale fading channel model as follows and put the resulting distribution of these variations into the log-normal part in the formula.

$$rx(d) = Po - 10n\log_{10}\frac{d}{do} + N$$

Highway Mobility Model:

The moment of the vehicles on highways depends on the high speeds of all vehicles, velocity, traffic density, the weather conditions, and the behavior of the drivers. So, by using these two models, we can achieve more accurate and reliable link between vehicles.

A macroscopic traffic flow model is a mathematical model that expresses the traffic flow characteristics like flow, density, mean speed of a traffic stream, etc. while the microscopic traffic flow models feign single vehicle-driver units, so the microscopic properties such as position and velocity of a particular vehicles is represented by dynamic variables of the models. The position of each single vehicle is needed to find that whether two vehicles come in the range of each other so that they can communicate with each other. By using the transmission power and the channel state information we can estimate the range of communication, and also the link reliability by the position and the velocity of a single vehicle.

2) *Route reliability definition:*

The link reliability is defined as the stable duration of the communication link between two vehicles. Link expiration time (LET) denotes the maximum time lasts from establishing the link to one vehicle moving out from the communication range. To calculate LET, speed and position of each vehicle are needed.

Consider we have two nodes i and j to calculate LET of them. Let R is the transmission range of every node. Distance between them is |di,j| and velocity of each node is vi and vj. If this node moves in the same direction then, we have:

$$LET = \frac{R+ \alpha*|di,j|}{|vi-vj|} \tag{1}$$

If two nodes are moving towards each other, means they first come closer and then go far for this condition α is +1. However, if two nodes are not in the same direction means they never come closer, so there α is -1. If the nodes move in the different direction, we have the following equation.

$$LET = \frac{R+ \alpha*|di,j|}{|vi+vj|} \tag{2}$$

We can also calculate RET (Route Expiration Time) after calculating of LET. RET for a route is minimum LETs that make that route:

$$\text{RET} = \min \{\text{LET1, LET2, ... LETn}\} \tag{3}$$

Broken link is the link which break while communication. If route has less broken link then the route is more stable. Otherwise, due to high broken link there is more exchange control packet and more packet loss.

B. Proposed System:

Here we are going to propose a medium access control (MAC) protocol for ad hoc wireless networks that utilizes multiple channels with dynamism to give better performance [18]. This multiple channels are available at the physical layer of the network. The IEEE 802.11 standard allows the use of multiple channels, but the MAC protocol is designed only for a single channel.

Designing a MAC protocol that works with the multiple channels is not an easy, as many of current devices of IEEE 802.11 have one half-duplex transceiver. This transceiver can switch the channels dynamically, but it can only communicate in one direction on single channel at a time. Thus, when a host is attending on a particular channel, it cannot take part in the communication going on a different channel. Because of this, a new type of hidden terminal problem arises in this multi-channel environment, which we can call as multi-channel hidden terminal problem. So, this single-channel MAC protocol does not work properly in a multi-channel environment where nodes may dynamically switch channels.

The scheme increases network throughput significantly, although when the network is very congested. By using multiple channels, we can get a higher network throughput than using one channel, because by multiple channels there is a multiple transmission can be done without interruption of other. So, we are going to propose a MAC protocol which can enable the hosts to dynamically change the channels such that multiple communications can be done in the same region at the same time, in different channel. As we are going to work with an ad hoc network that does not depend on infrastructure, so there is no central ability to perform management of different channels. In our work the main idea is that, we have to divide the time into fixed-time intervals by using beacons, and also having a small window at the starting position of each interval to get information about the traffic and accordingly change channels for use during the interval.

To improve the throughput there are other several MAC protocols are also proposed. But due to multiple channels they require multiple transceivers per host or they not able to overcome from the multi-channel hidden terminal problem, which does not give good performance. This is the protocol that needs only one transceiver per host, but still it can solve the hidden terminal problem in a multi-channel environment. As the protocol requires one transceiver per host, we are able to implement it with a hardware complexity which is comparable to IEEE 802.11.

TABLE 1: COMPARISON TABLE FOR BOTH THE SYSTEMS:

Sir no.	Existing System	Proposed System
1.	This system works with the LET concept to find the reliable route.	This system is the extended work of the existing system with multichannel concept.
2.	Single channel is present between nodes.	Multiple channels are assigning between nodes.
3.	Routes selected based on metrics like hop distance, signal strength, degree of stability and expected transmission time.	Load balancing between available channels is done
4.	There is no such guarantee.	Guarantees route establishment if the route can be established in a single channel network with same topology.
5	It gives moderate result as compare to new proposed system.	As having multiple channels its throughput will increase and give more efficient result with respect to our performance metrics.

IV. Simulation Setup and Results:

In this section first we are considering the performance metrics considering which the enhancement is done. After it the description of the simulation environment and the first part of the expected outcome is presented. It has the results of the comparison between the traditional AODV routing protocol and the AODV-L routing protocol by using simulator NS2.34.

A. Performance Metrics:

Packet Delivery Ratio: Packet delivery ratio is a very important factor to measure the performance of protocol in any network. Packet delivery ratio is the ratio of number of packets received at the destination node to the number of packets sent from the source node. The performance is better when packet delivery ratio is high. Mathematically it can be shown as equation.

Loss Packet Ratio: Packet Loss Ratio is the ratio of the number of packets that never reached the destination to the number of packets originated by the source. Mathematically it can be shown as equation.

Route Stability: The link reliability is defined as the stable duration of the link between two vehicles.

B. Simulation Environment:

Here we are comparing the above mentioned three performance metrics. Our simulation has been done in a 1000m×1000m area. Vehicles are randomly placed on the road and go straightly ahead in high velocity until they reach intersection. The static nodes which are the roadside unit do not move. The simulation experiment is conducted on NS2.34 and IEEE 802.11. The transmission rate is of 2Mbps and transmission range s of 200m used as MAC protocol. The number of vehicles considered is 30-50. Here the source node is the UDP agent and Destination node is Null agent communicates using 512 CBR (Constant Bit Rate).

TABLE I. COMPARED PERFORMACE METRICS

Parameter	Value
Network Simulator	NS2.34
Simulation Area	1000 x1000 m
CBR	512 bytes/sec
802.11 rate	2 Mbps
Transmission Range	200 m
No. of vehicles	30-50
No. of roadside units	8-10
Simulation time	500-1000 sec

Based on the simulation result we have generated the graph which shows the performance differences between AODV and AODV-L. The graphs are given below. These graphs are generated for 30 nodes but at the different velocity (20, 40, 60, 80, 100 ms). Figure 3 is of the Packet Delivery Ratio at different velocities shows that the pdr of AODV-L is more than AODV. Figure 4 is of the packet loss ratio shows that plr of AODV-L is less than AODV. Figure 5 is for the route reliability of these protocols of which AODV-L give more efficient result than AODV routing protocol.

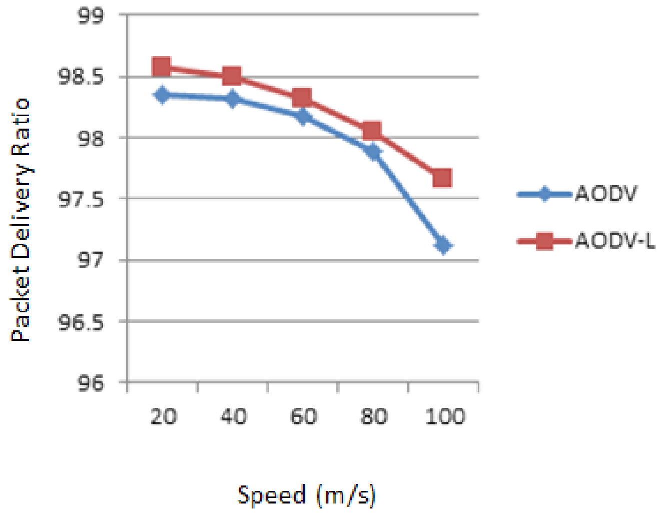


Fig 3. Packet delivery ratio.

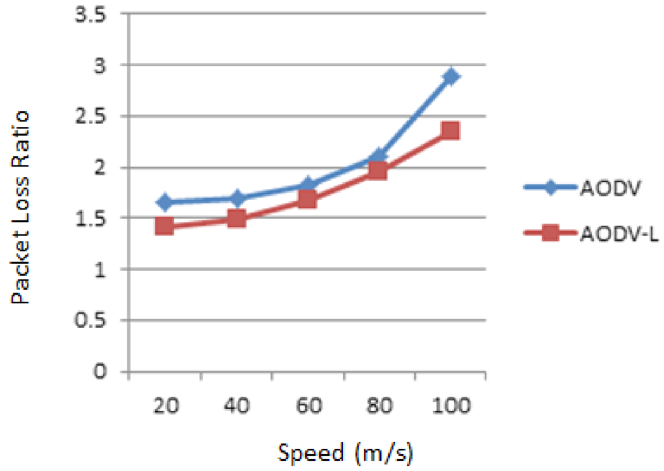


Fig 4. Packet loss ratio.

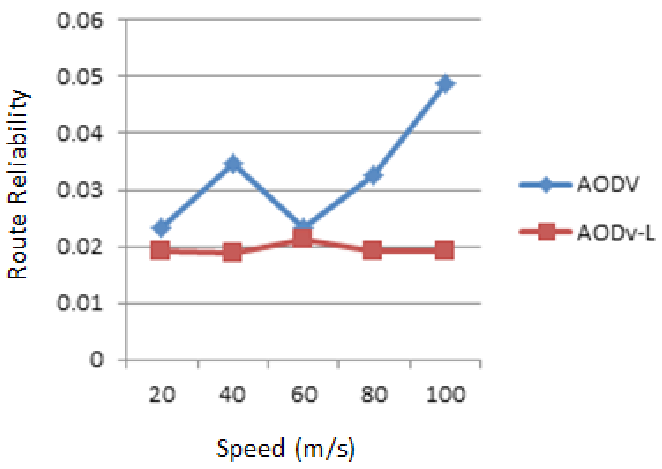


Fig 5. Route reliability.

V. Conclusion

In this paper, firstly, we introduce about the existing system which that enhances the stability and reliability of the routing protocol in VANETs. The idea behind the offered scheme AODV-L is the Highway Mobility Model and the Stochastic Large- Scale Fading Model with the applied concept of calculation of LET, which is the strategy for the selection of stable routes. This protocol gives the effective Simulation results of this part of the work in terms of packet delivery ratio, loss packet ratio and route reliability. In the future work, we will carry out more intensive simulation by adding the concept of multichannel MAC protocol to this AODV-L routing protocol. This scheme will give more effective output in terms of throughput and also the considered parameters.

Acknowledgment

I would like to express my gratitude towards my mentor Ms. Rashmi Welekar, Assistant Professor in Department of Computer Science and Engineering from Ramdeobaba College of engineering and management, Nagpur for her guidance and encouragement which helped me to understand such a vast topic of Vehicular Ad-hoc Networks. I would like to thank my family as well, for their continuous support and encouragement.

References

[1] Yang He, Wenjun Xu and Xuehong Lin, "A stable routing protocol for highway mobility over vehicular ad-hoc networks", IEEE 2015.

[2] G. Karagiannis, O. Altintas, E. Ekici, *et al.*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Commun. Surveys & Tutorials*, vol. 13, no. 4, pp. 584–616, 2011.

[3] M. H. Eiza, Q. Ni, T. Owens, *et al.*,"Investigation of routing reliability of vehicular ad hoc networks," *EURASIP journal on wireless communications and networking*, vol. 1, pp. 1–15, 2013.

[4] H. Guo, F. Tao, M. Ma, *et al.*, "A reliable route selecting algorithm for vehicle communication," *2009 IEEE 7th International Conference on Information, Communications and Signal Processing (ICICS)*, 2009, PP. 1–5.

[5] S. R. Das, E. M. Belding-Royer, and C. E. Perkins, "Ad hoc on-demand distance vector(AODV) routing," 2003.

[6] N. Akhtar, O. Ozkasap, and S. C. Ergen, "VANET topology characteristics under realistic mobility and channel models," *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, 2013, pp. 1774–1779.

[7] Z. Tang and J. J. Garcia-Luna-Aceves, “Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks,” in *Proc. of IEEE INFOCOM*, 1999.

[8] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, “A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks,” in *Proc. of IEEE INFOCOM*, 2001.

[9] A. Nasipuri, J. Zhuang and S. R. Das, “A Multichannel CSMA MAC Protocol for Multihop Wireless Networks,” in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, September 1999.

[10] A. Nasipuri and S. R. Das, “Multichannel CSMA with Signal Power-based Channel Selection for Multihop Wireless Networks,” september2010.

[11] N. Akhtar, O. Ozkasap, and S. C. Ergen, “VANET topology characteristics under realistic mobility and channel models,” *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, 2013, pp. 1774–1779.

[12] V. Namboodiri and L. Gao, “Prediction-based routing for vehicular ad hoc networks,” *IEEE Trans. Veh. Tech.*, vol. 56, no. 4, pp. 2332–2345, 2007.

[13] L. Cheng, B. E. Henty, D. D. Stancil, *et al.*, “Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1501–1516, 2007.

[14] O. Renaudin, V. Kolmonen, P. Vainikainen, *et al.*, “Non-stationary narrowband MIMO inter-vehicle channel characterization in the 5-GHz band,” *IEEE Trans. Veh. Tech.*, vol. 59, no. 4, pp. 2007–2015, 2010.

[15] N. Jain and S. Das, “A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks,” in *Proc. of the 9th Int. Conf. on Computer Communications and Networks (IC3N)*, October 2001.

[16] F. A. Tobagi and L. Kleinrock, “Packet Switching in Radio Channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy tone solution,” *IEEE Transactions on Communications, COM-23*, 1975.

[17] A. Nasipuri, S. Ye, J. You and R. Hiromoto, “A MAC Protocol for Mobile Ad Hoc Networks using Directional Antennas,” in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, September 2000.

[18] J. So and Nitin Vaidya,” Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver”, May 2004

Divya Rathi she is a student of Master of Technology in Computer science engineering at Ramdeobaba college of Engineering and Management, Nagpur. She has received her Bachelor of Technology degree in Computer Science from Sant Gadagebaba University, Amravati. Her area of interest is networking and she published one survey paper in International conference named RICE 2016.

Rashmi R. Welekar, she is working as Assistant Professor at Ramdeobaba college of Engineering and Management, Nagpur. She has been teaching from last 10 years in the field of Computer Science & engineering. She did his M. Tech from Nagpur University. Her areas of interest are Image Processing, Networking, and Pattern Recognition. She has about 10 research papers published in International Journals, 6 international conferences and 1 national conference.

Conceptual Model for Smart Cities: Irrigation and Highway Lamps using IoT

Vijender Kumar Solanki, M.Venkatesan, Somesh Katiyar

Research Scholar, Anna University, Chennai, India

***Abstract* — Keeping in mind the need to preserve energy as well as utilize the available at its best the need was felt to develop a module that would be able to sort out the problem where resources such as water and electricity were wasted, in urban as well as rural area. Resource (electricity) was wasted as beside the point operation of Highway & High Mast Lamp; while wastage of water followed by improper trends and methodologies imparted for watering of city park, road side plantation and highway plantation. Thus as per Energy survey statistics of a City (Lucknow, India) it was found that major portion of resources (water and electricity) were being wasted due to negligent activities of officials who were in charge of resource management. So to facilitate energy saving trends and to completely modernize it to autonomous system, module below is proposed which incorporates modern technological peripheral and has its base ingrained in IoT (Internet of Things) which when put into consideration would result in large scale resource and energy saving.This developed module incorporates the peripherals such as Arduino, Texas Instruments ultra low power kits etc. in accordance with software technology including Lab View which help to monitor as well as control the various operation from the base station, located far away from the site. Lab View Interface interacts with all the module located at various city parks, subways and highway lighting modules. Later below in several section a detailed pattern and application frame has been put up.**

***Keywords* — Smart City, Arduino, Lab-View, Automatic Irrigation System, (Highway lamp / High Mast Lighting) Operation and Control.**

I. Introduction

As per the growing rate of population with spontaneous consumption of resources, creates in the need for the managing the available resources at its best. So a need was felt to manage the outflow of the two major resources i.e. water and electricity and to formulate out, that's how it can be protected from getting wasted and could be utilized at its best. [1]

As during the survey study, it was found that (Lucknow, in India) the practices were manual and a major portion of resources was wasted due to slothful and unconcerned behavior leading to plant death and unwanted operation of the lights.

So using modern technology, and statistical, survey based study it was found that that major portion of the resources (water and Electricity) could be managed out and preserved by managing their controlled flow in an allocated area/city/state/territory with channelized Irrigation system and employing modern means for control of Highway lamps and High Mast Lighting. [2,3]

- Primarily, this module would be capable to help, rule out the problem faced with irrigation process, which was carried out manually, and improper trends were practiced which were either resulting in resource (water) wastage or when not followed properly resulting in plant dying out indirectly unfavorable habitat.
- Secondarily the module helps in controlling the Lamps of Various Highways based on collection of data from weather forecast report (such as visibility, mist, fog etc.) and toll plaza as per the traffic density so that accordingly the lights/lamp/ high mast lamp could be operated.

II. Technology Adopted

The developed module, thus incorporates the solution for both of the issues:

I.) Advanced Irrigation System for Parks and Road Side Plantation: It includes grouping together of various peripherals together using IoT which help in:

- Data accusation (such as: Status of fountain Running/Idle position; Water Level in the Tank; Soil Moisture Content: rated b/w 0 to 100) [4,5,6]
- Remote Operation Enables to control the various operations such as Operation of Lights, Operation of Fountain Pump, Operation of Sprinklers of City Parks and Road Side Plantation.

II.) Advanced Highway and High Mast Lighting System: Provides automatic control of the lights of the Highway and High Mast Light based on the:

- Weather Forecast (Visibility, Fog etc.): As it was found that during bad weather less visibility, fog affected condition it is necessary to operate all the lights at its full.
- Traffic Density: Data from the Toll Plaza has to be collected so accordingly the operation of Lamps if necessary could be operated in the available modes. (Alternate Mode, Full Mode operations)

III. Problem Formulation

The module at base station includes the Lab-View platform installed PC enabling:

- Highway lamps to be controlled as per the requirement through remote access [7]
- Irrigation related functioning such as: Water Level in Tank, Operation of Sprinkler, and Operation of Fountain Lights can be achieved [8].

In the schematic figure below Fig.1 represents the base station in continuous communication link with the discrete module i.e. Highway / High Mast Lamp and Irrigation module with on sites comm. through 30 ft Rx Tx Weatherproof Communication Link Network (SPN2dp8 for 5Km radii Communication range with 0 obst.)

At center lies the base station where Lab view platform based Host PC is installed while on to left half depicts the Highway and high mast Lighting control module where Arduino set's connected to

DOI: 10.9781/ijimai.2017.435

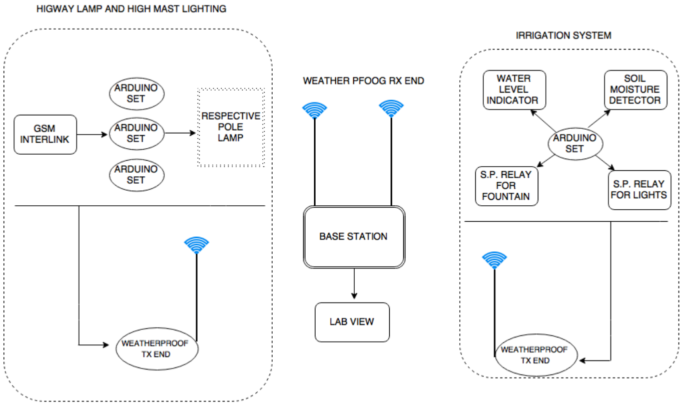


Fig. 1. Working Module describing the concept of interlink network and peripherals of Advanced Highway and Irrigation System in Smart City

sensor for feedback are inter-connected through GSM link via (SIM com900A module for Indian telecom) and giving resultant feed up to weatherproof TX. Whereas on to right-half irrigation module with sub-discrete peripheral such as Water level Indicator, Soil moisture detection, Fountain operator, and fountain light operator connected for data exchange to Arduino mega 2560 and then to weather proof TX end to base Rx end.

IV. Advanced Highway and High Mast Lighting

Problem Observed: Over the course of time it was observed that much of the energy was wasted as these highly energy rated lamps remained to "ON" state, being in operation, although there was no requirement, this was the result of negligence of officials/engineers in-charge of controlling the operation.

Fig 2. A layout of current employed system as per for installation of High Mast Lighting and Highway lamps.

Thus as a measure of substitute to existing system (as solution) module is proposed that is completely automatic and highly efficient. This would help to prevent the losses, will help in better saving and optimal utilization of resources. [9,10,11]. Fig. 2 depicts current trends of a city installed with manually controlled Highway and High Mast Lights. Fig3. Presents IOT based solution using Lab view as a running base platform to manage the lighting system of Highways as per the following aspect into consideration:

- User end Software Controlled
  - Visibility on the road
  - Traffic density on the road

    In this at base station a trained operator or engineer controls the light as per stats and data from live footages as per the traffic density ratio.
  - Autonomous via software control

In this mode data from servers of toll plaza and weather forecast are considered and as per initialized threshold value the respective light are operated, complete activity being autonomous.

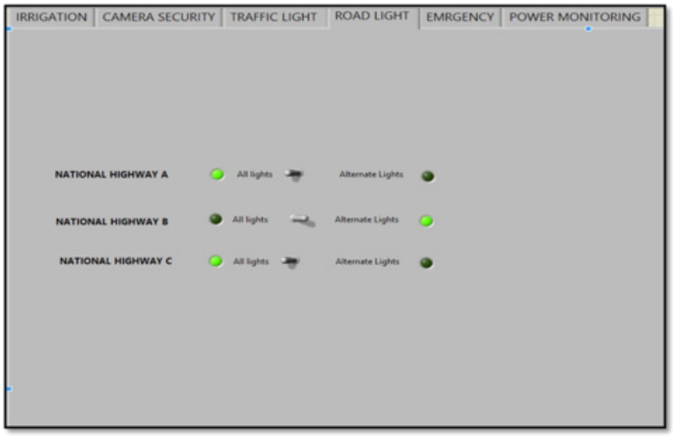


Fig 3. Lab View VI Module for Controlling of Highway Lamp's (can be controlled Manually through switching from Base Station as per the data received from Toll Plaza , and weather forecast report) (simulation based result).

V. Operational Framework

Requirement:

- Data from the Toll Plaza [12,13]

  Data from the toll plaza is collected through server based channels at the Smart City and Security base station, where a systematic study is carried out about the traffic density on that particular Highway.

- Data from weather Forecast[14]

  Weather forecast data is required for the keeping in the record of the weather as well as the visibility, that depends on rain, fog, mist etc.

- Arduino Kit:

  Build type: Arduino UNO

  Purpose: Receive the signal form sensors such as Moisture detector or water level indicator and transfer the data to GSM module for transmission to Smart City baseStation [15,16,17]

  Work Type: It performs both digital as well as analog collection of data and develop digital codes and results that can be manipulated and transferred or recorded.

- Weatherproof 300 ft. TX. Kit for video Link [18,19]

  Purpose: Long range Video System, Birthing and Livestock Footage (transmission Type)

  Work Type: (a.) Real time footage capture (b.) Digitalization (Encoding) for Easy Transmission

- Weatherproof 300 ft. RX. Kit for video Link [20]

  Purpose: Long range Video System, Birthing and Livestock Footage (Receiving Type)

  Work Type: (a.) Real time footage, receiving (b.) Decoding of receiving Data

VI. Irrigation System

In order proper maintain lush green plants / trees in smart city have to be watered regularly, i.e. watering of fields of city parks and road side plants. Earlier practices involved either water channeling or manual irrigation, thus when proper care is not undertaken or not properly practiced leading to plant death. Apart from this manually operating water pumps for tank filling and sprinkler operation led to water as well as electrical energy wastage as beyond the need operation. [21]

This IOT based module enables to maintain the well suited condition for the plants so that, they can grow at best to provide in shed and also maintain the natural beauty. [22,23]

The moisture content in the soil can easily be measured using the Soil Hygrometer Detection Module, which is buried in the soil and gives the continuous reading. As per the plant requirement the operation of sprinkler can be operated manually or automatically obtain the required results and maintain the best suitable condition for growth of plants. And for maintaining proper level of water in water tanks water level indicators are being employed to maintain the subsequent water for irrigation purpose. [24,25]

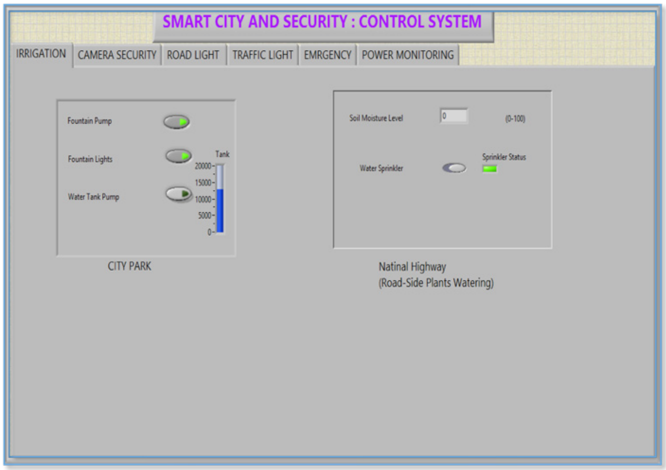


Fig 4:Lab View VI Module for Advanced Irrigation System help to control as well as for Data Accusation of Water Tank Level, Operate Sprinkler, Fountain and Fountain Lights.

Fig. 4 depicts the Lab View module for various condition such as:

- Control and status of water sprinkler
- Soil moisture level at time of sprinkler operation (rated 1% - 100 %)
- Control and status of water pump
- Control and status of fountain pump
- Control and status of fountain lights
- Amount of water level in heavy capacity water tank

VII. Operational Framework

Requirement:

- Soil Moisture Detector:

  Build Type: PIC Atmel Based

  Purpose: To collect the data about the moisture content of the soil

  Work Type: The results are obtained in form of resistance, between two electrodes giving the conductivity

  lesser the moisture: more the resistance

  more the moisture: more the conductivity

- Arduino Kit:

  Build type: Arduino UNO

  Purpose: Receive the signal form sensors such as Moisture detector or water level indicator and transfer the data to GSM module for transmission to Smart City base Station

  Work Type: It performs both digital as well as analog collection of data and develop digital codes and results that can be manipulated and transferred or recorded.

  Ñ Texas Instruments MSP430G2553:

  Purpose: Ultra low power with booster pack plug in module, mainly for frequency decoding, serial interport , relay board power, segment display

- GSM Kit: (Rx and Tx)

  Build Type: SIM 900 Module for Data TX and RX

  Purpose: The digital data received form the Arduino has to be transmitted to base station of the Smart City and Security to operate the sprinkler located

- Water Level indicator:

  Build Type: SD512 Resistive (Non Corrosive)

  Purpose: Water level is also a resistive type basically modeled to carry out and note down the capacity of the water tank.

- Relay Single Pole:

  Build Type: JQC3F 5 Pin SPDT

  Purpose: Controlling and Switching of Fountain Lights and Sprinkler

Fig 5. Depicts the simulation based operation of various pump set with indicator repressing overflow state and power status. In case of detection of high signal value either at over flow or at moisture level beyond threshold operation the relay card in Fig 6. Carries out the necessary operation i.e. to bring back the pump to OFF state. In other words, the relay card can be regarded as main functioning unit for operation of heavy motors. Below is simulation based study carried out with single relay to drive up a single motor. While implementing this circuit in practical use repeated number of relay cards have to be connected to drive each motor for desired purpose.

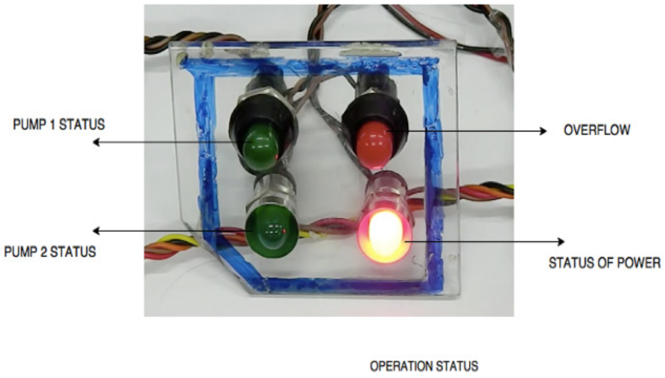


Fig 5:Operation Status display (Real Time) installed at operating end.

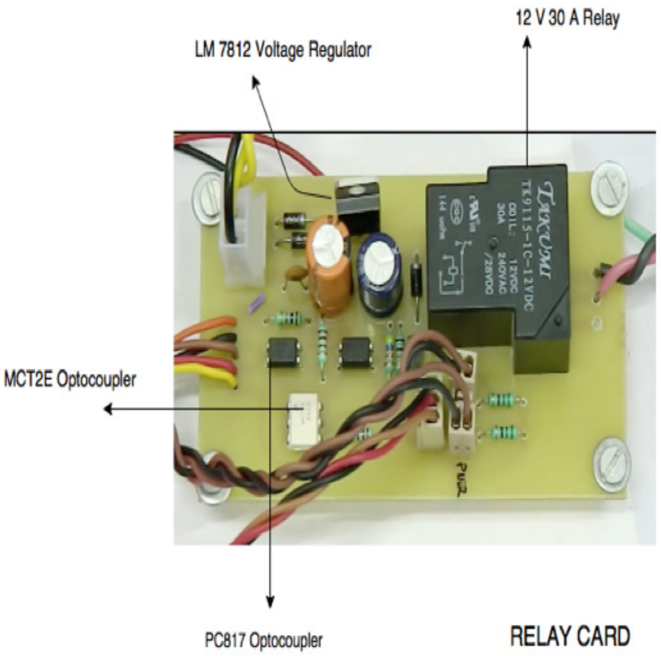


Fig. 6:Relay card for Driving Motor: for water pump set, and sprinkler operation.

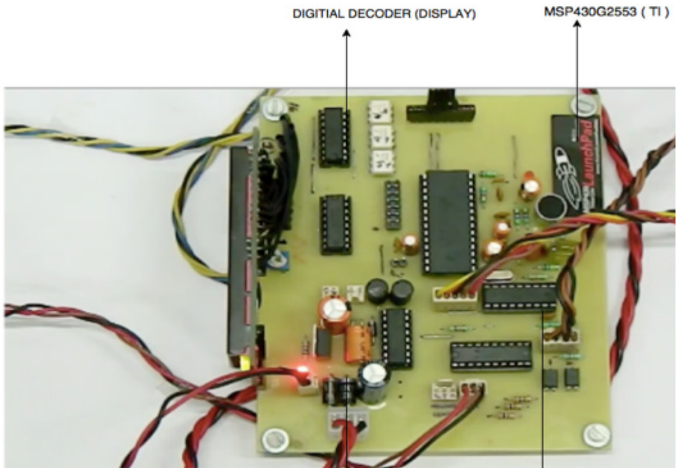


Fig. 7: Master Card driver for Serial channel interface, relay card driving and peripheral operation.

While the min processing units where received signals from base station are received is decoded at master card shown in Fig. 7 which carries out necessary calculation as well as decoding of received data and serves the signal to relay card for specific motor operation.

VIII. Future Work

The developed module has been tested on simulation and works well with android platform. But currently its pejorative to iOS platform due to intricacy involved. Apart from this future work is more focused toward maintaining the accuracy and precision for a particular task to be carried up. Such as detecting moisture in soil is limited because of limited analog pins in Arduino so future work would incorporate analogous sensor data collection and working on precision and accuracy.

The major achievement could be attained by modulating the signal frequency over a single band for high bit-rate data communication, using advanced data communication devices.

IX. Result and Conclusion

- After installation of the module to city better controllability of sprinkler's, tracking level of water and simultaneously operate the water pump could be successfully achieved which are being tested, and thus resources both water and electricity could be managed and saved unto a great extent.
- Apart from this major portion of electrical energy which was initially wasted as beyond the point operation of Highway lamps and High mast lamps would now be avoided and better safe and saving operation could be carried out.
- Data form both the sources are gathered up and studied both manually and technically and according to the comfort the lights can be operated.
- Table below depicts the study of operation of Highway lamps (Lab View VI software based simulation result) for Comparative study of the saving made after installation. The trends and values are as per weather condition (in Lucknow) 26.30 : 27.10 North latitude : : 80.30 : 81.13 East longitude (India). As per day pattern and natural light availability.

TABLE I Study of Operation Highway Lamps

	OLD Installation	Modern Installation	Savings
Jan	13 Hrs.	9 Hrs.	4 Hrs.
Feb	12.5 Hrs.	8 Hrs.	4.5 Hrs.
May	10 Hrs.	7 Hrs.	3 Hrs.
July	9 Hrs.	7 Hrs.	2 Hrs.
Sep	8 Hrs.	7 Hrs.	1 Hrs.
Nov	9 Hrs.	8 Hrs.	1 Hrs.

The graphical statics of the above Table 1 is Chart 1 which helps to comprehend it better.

Chart 1: Stats showing the difference if energy consumption or in words savings made after installation of the module, data to related chart form Table 1.

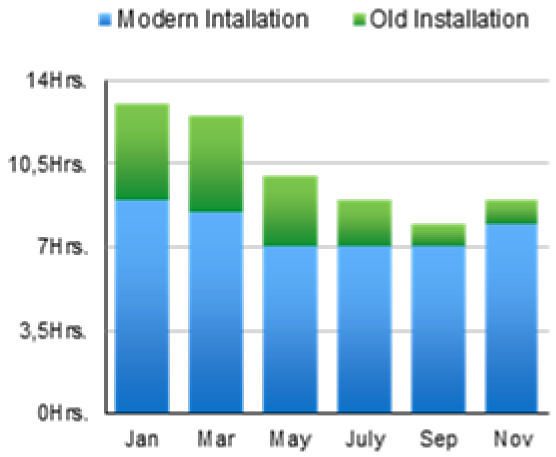


Table 2 here shows an outline of appliances installed, their wattage rating, and approx. consumption of Electric Power in a single day.

TABLE II. OUTLINE OF APPLIANCES INSTALLED

Appliance	Wattage rating(kW)	Installation (No.)	Hrs. of Operation (Hrs)	Power Consumption (kWh)
Air Conditioner	3.3	1	6 *Conditional	3.3 *6 =19.8
Ceiling Fan	0.73	5	8 *Conditional	0.73*5*8 =17.2
Florescent Lamp	0.4	7	4 *Conditional)	0.4*28 =11.2
Geyser	1.2	1	-/ **Occasional	- -
Outdoor Lamp	0.80	2	4 *Conditional	0.8*8 =6.4
Garden Lamp	0.6	8	-/ ***Optional	- -
Garden Sprinkler	0.43	2	2 ***Optional	0.43*4 =1.72
Desktop Computer	0.145	1	8 *Conditional	0.145*8 =1.16
Fridge	0.433	1	9 *variable	0.433*9 =3.897
Total:	-			61.377 Units

Hrs. of Operation:

*Conditional: Appliance operation may vary, example is just mend to illustrate

**Occasional: These appliances are not that frequent as others

***Optional: These appliances/equipment's are not commonly installed in every home

Table 2 help us to comprehend that the saving made from data of table 1 and help to drive loads of domestic utility.

Illustrative Example: *Combining the data from Table 1 and Table 2 it can easily be comprehended that let say for a day in Jan per day saving made is of 4 Hrs. Now,*

Taking case of 2 Km road having 22 Road Lamps of 500W (Model :Havells LHSH10050099) installed

So,

*No. of Lamps * Wattage Rating * No. Hrs of Operation = Power Consumption in Units*

$$N * W * Hr = kWhr\ (Units)$$

O_l *22 * 0.500W * 13 = 143 kWh {old Installation* O_l*}*

M_o *22 * 0.500W * 9 = 99 kWh {modern installation* M_o*}*

$O_l – M_o$ *= 143-99 = 44kWh{saving made}*

: Since in table 2 on an avg. a domestic appliance utilizes (61.377 ~ 62) units per day from which 44 units can be supplied from savings made and thus their remains only 18 units to be met.

Therefore, from above statics, savings made from operation Highway lamp and High mast lamp can be studied and compared to per day domestic utility to frame out savings. Thus if this system brought up and is employed it would definitely help to make enormous savings and serve out domestic users.

Acknowledgment

The authors are extremely grateful to the reviewers whose comments have led to significant improvement in the quality of the paper.

References

[1] Vijender Kumar .Solanki, M. Venkatesan, S. Katiyar, Vijay B. Semwal, P. Dewan, N. Dey 2016, 'Advanced Automated Module for Smart and Secure City', Elsevier: Procedia Computer Science, Vol. 78, Page 367-374.

[2] A. Khattak, M. Pervez, Z. Jehad Sarkar, A. M. and Y. Lee, "Service Level Semantic Interoperability", *10th IEEE/IPSJ International Symposium On Applications And The Internet, Saint*, pp. 387-390.

[3] P. Barnaghi, W. Wang, C. Henson and K. Taylor, "Semantics for the Internet of Things: Early Progress and Back to the Future", *International Journal on Semantic Web and Information Systems*, vol. 8, no. 1, 2012

[4] L. Atzori, A. Iera and G. Morabito, "The internet of things: A survey", *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, 2010.

[5] M. Dohler, I. Vilajosana, X. Vilajosana and J. LLosa, *Barcelona Smart Cities Congress 2011*.

[6] G. Flouris, D. Plexousakis and G Antoniou, "A Classification of Ontology Change", *The Poster Session of Semantic Web Applications and Perspectives (SWAP), 3rd Italian Semantic Web Workshop*

[7] N. Bressan, L. Bazzaco, N. Bui, P. Casari, L. Vangelista and M. Zorzi, "The deployment of a Smart Monitoring System using Wireless Sensor and Actuator Networks", *Proc. of IEEE Smart Grid Comm 2010*.

[8] C. E. A. Mulligan and M. Olsson, "Architectural implications of smart city business models: an evolutionary perspective", *IEEE Communications Magazine*, vol. 51, no. 6, pp. 80-85, 2013.

[9] N. Walravens and P. Ballon, "Platform business models IoT-A consortium", *Mission-IoT-A: Internet of Things Architecture*.

[10] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson and A. Oliveira, "Smart Cities and the Future Internet: Towar ds Cooperation Frameworks for Open Innovation", *The Future Internet, Lecture Notes in Computer Science*, vol. 6656, pp. 431-446, 2011.

[11] D. Steinberg and S. Cheshire, "Zero Configuration Networking: The Definitive Guide", *O'Relly Media, Inc.*, 2005.

[12] A. Narayanan and V. Shmatikov, "Robust De-anonymization of Large Sparse Datasats", *2008 IEEE Symposium on Security and Privacy (sp 2008). IEEE*, pp. 111-125.

[13] Hussain M.J. Almohri, Danfeng (Daphne) Yao and Dennis Kafura, "Process Authentication For High System Assurance", *Ieee Transactions On Dependable And Secure Computing*, vol. 11, no. 2, 2014.

[14] Lin Gu, Deze Zeng, Peng Li and Song Guo, "Cost Minimization For Big Data Processing In Geo-Distributed Data Centers", *10 March 2014; Date Of Current Version 30 October 2014. Digital Object Identi_Er 10.1109/ Tetc.2014.2310456*.

[15] A. Verma, L. Cherkasova and R. Campbell, "Aria: automatic resource inference and allocation for mapreduce environments", *Proc. ACM ICAC*.

[16] U. Kang, C. E. Tsourakakis and C. Faloutsos, "PEGASUS: mining peta-scale graphs", *Knowledge and Infomation Systems*, vol. 27, no. 2, 2011.

[17] Jeffrey Dean and Sanjay Ghemawat, "Mapreduce: Simpli_Ed Data Processing On Large Clusters", *At To Appear In Osdi*, 2004.

[18] Brian Hellig, Stephen Turner, Rich Collier and Long Zheng, "Beyond Map Reduce: The Next Generation Of Big Data Analytics", *Solving Big Data Problems Hamr-Eti.Com Rcollier @ Etinternational.Com 302.*

[19] B. CliordNeuman, "Proxy-Based Authorization and Accounting for Distributed Systems", *the 13th International Conference on Distributed Computing Systems*.

[20] F. Canan, PembeMuhtaroglu and SenizDemir, "Business Model Canvas Perspective on Big Data Applications", *2013 IEEE International Congress on Big Data*

[21] Automation of an irrigation system. Available at: http://www.thefreedictionary.com/irrigation+ditch

[22] Agriculture system in Bangladesh. Available at: http://www.bookrags.com/ history/terrace-irrigation-ema-05/

[23] Drip Irrigation system. Available at: http://en.wikipedia.org/wiki/Drip-irrigation

[24] South Carolina Irrigation / Irrigation equipment. Available at: http://www.clemson.edu/irrig/Equip/Trav.htm

[25] Dr. B.A.A. Mustafi, Dr. Md. Rafiqul Islam: Development of agricultural politics in Bangladesh. Available at: http://www.ipipotash.org/udocs/Development-of-agricultural-policies-in-Bangladesh-MAA-Mustafi.pdf

Vijender Kumar Solanki received the Master in Computer Application and Master in Engineering degree (Computer Science & Engineering) from Maharishi Dayanand University, Rohtak, Haryana, India.(2004 & 2007). He is pursuing Ph.D. (Computer Science & Engineering) from Anna University, Chennai, Tamilnadu, India. His primary research interests are in Network Security, Smart Cities and Big Data. He is reviewer of IEEE, Springer & Elsevier conferences and many International journals. He was the Guest Editor of IJRSDA, Spl Issue on "RICE". He has delivered many Lectures in FDP, Workshop and conferences. He can be contacted at spesinfo@yahoo.com

M. Venkaesan is professor in KSRIET, Department of computer science and engineering. He is M.E & Phd in computer science engineering. His area of research is WSN, Security, Big Data and Smart city. He is having more than decade experience in teaching and research. He is affiliated and approved supervisor with Anna university Chennai. He is reviewer of numerous high indexed journals. He has conducted many FDP, seminar and conferences successful with high attendances of delegates from in and around country. he can be contact at venkatesh.muthusamy@gmail.com

Somesh Katiyar born on Dec 1994, Lucknow India. Received B. Tech degree in Electrical & Electronics from Chandigarh Group of College, PTU Punjab – India in 2015. His research interest lies in sensor, microcontrollers, machine learning, smart cities , Internet of Things. E-mail : someshkatiyar99@gmail.com

Migrating C/C++ Software to Mobile Platforms in the ADM Context

Liliana Martinez[1], Claudia Pereira[1], Liliana Favre[1,2]

[1] Universidad Nacional del Centro de la Provincia de Buenos Aires, Tandil, Argentina
[2] Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina

***Abstract* — Software technology is constantly evolving and therefore the development of applications requires adapting software components and applications in order to be aligned to new paradigms such as Pervasive Computing, Cloud Computing and Internet of Things. In particular, many desktop software components need to be migrated to mobile technologies. This migration faces many challenges due to the proliferation of different mobile platforms. Developers usually make applications tailored for each type of device expending time and effort. As a result, new programming languages are emerging to integrate the native behaviors of the different platforms targeted in development projects. In this direction, the Haxe language allows writing mobile applications that target all major mobile platforms. Novel technical frameworks for information integration and tool interoperability such as Architecture-Driven Modernization (ADM) proposed by the Object Management Group (OMG) can help to manage a huge diversity of mobile technologies. The Architecture-Driven Modernization Task Force (ADMTF) was formed to create specifications and promote industry consensus on the modernization of existing applications. In this work, we propose a migration process from C/C++ software to different mobile platforms that integrates ADM standards with Haxe. We exemplify the different steps of the process with a simple case study, the migration of "the Set of Mandelbrot" C++ application. The proposal was validated in Eclipse Modeling Framework considering that some of its tools and run-time environments are aligned with ADM standards.**

***Keywords* — Architecture-Driven Modernization, Haxe, Migration, Metamodeling, Mobile Platform, Model-Driven Development**

I. Introduction

Today, mobile phones are the most used computing platform worldwide. The wide spread use of mobile computing, that emerged from the integration of cellular technology with the Web, has contributed to opening up opportunities for new paradigms such as Pervasive Computing, Cloud Computing and Internet of Things (IoT).

Pervasive Computing, also called Ubiquitous Computing is the idea that almost any device can be embedded with chips to connect the device to a network of other devices. The goal of pervasive computing, which combines current network technologies with wireless computing, voice recognition and Internet capability, is to create an environment where the connectivity of devices is unobtrusive and always available. Smartphones come with a variety of sensors (GPS, accelerometer, digital compass, microphone, and camera), enabling a wide range of mobile applications in Pervasive Computing.

Cloud Computing is an Internet-based computing for enabling ubiquitous, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly supplied with minimal management effort. This generates enormous amount of data, which have to be stored, processed and accessed. Cloud computing has long been recognized as a paradigm for Big Data storage and analytics providing computing and data resources in a dynamic and pay-per use model. Mobile Cloud Computing is the combination of Cloud Computing, Mobile Computing and Wireless Network to provide computational resources to mobile users, network operators, as well as cloud computing providers.

There is no single universal definition for Internet of Things. The Oxford Dictionaries defines IoT as "the interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data". Gartner defines the Internet of Things (IoT) as "the network of physical objects that contain embedded technology to communicate, sense or interact with their internal states or the external environment" [1]. This can generate volumes of real-time data that can be used by enterprises for a variety of business applications. The IoT is becoming so pervasive and several studies predict that will be more than 30 billion IP-connected devices and sensors in the world by 2020.

Connectivity is central in Internet of Things. IoT extends Internet connectivity beyond traditional mobile devices to a diverse range of devices and everyday things that utilize embedded technology to communicate and interact with the external environment, all via Internet. Every object is integrated to interact with each other, allowing for communications between objects, as well as between human and objects, and the control of intelligent systems.

Pervasive computing, Cloud Computing and IoT face similar problems and challenges and smartphones have been one of the greatest facilitators of them. They are pursuing similar use cases, including smart cities, environmental monitoring, agriculture and home automation, and health and monitoring. These technologies will evolve and merge into only one following the vision of Mark Weiser: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it" [2]. IoT hardly could exist without the existence of Cloud Computing, as IoT need a network, storage, very cheap analytical possibilities to collect this information and analyze it in a meaningful way. IoT is also based on the same concept of the Pervasive Computing: having sensors and processors in everyday objects to determine their operation.

IoT is possible because thanks to mobile computing, the electronic miniaturization advances allow cutting-edge computing and communication technology to be added into very small objects. Besides mobile computing promoted the globalizations of 3G and 4G networks and today it is promoting 5G. Mobile Computing also facilitated the development of distributed processing to create a network of billions of devices. In summary, Mobile Computing is crucial to harnessing the potential of Pervasive Computing, Cloud Computing and IoT and, without the existence of smartphones these paradigms would not exist.

The development of applications aligned to these new paradigms requires adapting desktop software components to mobile platforms

DOI: 10.9781/ijimai.2017.436

facing many challenges due to the proliferation of different mobile platforms. The high cost and technical complexity of targeting development to a wide spectrum of platforms, has forced developers to make applications tailored for each type of device. New programming languages are thus emerging to integrate the native behaviors of the different platforms targeted in development projects. In this direction, the Haxe [3] language allows writing mobile applications that target all major mobile platforms, such as Android, iOS and BlackBerry, in a straightforward way.

Novel technical frameworks for information integration and tool interoperability such as the Model-Driven Development (MDD) can help to manage a huge diversity of mobile technologies [4]. MDD provides principles and techniques to represent software through models at different abstraction levels. A specific realization of MDD is the Model-Driven Architecture (MDA) proposed by the Object Management Group (OMG) [5]. The outstanding ideas behind MDA are separating the specification of the system functionality from its implementation on specific platforms, managing the software evolution from abstract models to implementations. The essence of MDA is Meta Object Facility (MOF), an OMG standard for defining metamodels that provides the ability to design and integrate semantically different languages such as general-purpose languages, domain specific languages and modeling languages in a unified way. Significant advantages can be made of this unification to construct powerful mobile design environments. The modeling concepts of MOF are classes, which model MOF meta-objects; associations, which model binary relations between meta-objects; Data Types, which model other data; and Packages, which modularize the models [6]. Consistency rules are attached to metamodel components by using OCL [7]. MOF provides two metamodels EMOF (Essential MOF) and CMOF (Complete MOF). EMOF favors simplicity of implementation over expressiveness. CMOF is a metamodel used to specify more sophisticated metamodels.

The Architecture-Driven Modernization (ADM) approach has established a set of solutions for information system modernization. ADM is defined as “the process of understand and evolve existing software assets for the purpose of software improvement, modifications, interoperability, refactoring, restructuring, reuse, porting, migration, translation, integration, service-oriented architecture deployment” [8]. The OMG ADM Task Force (ADMTF) is developing a set of standards (metamodels) to facilitate interoperability between modernization tools. To date, ADMTF has published the standards such as KDM (Knowledge Discovery Metamodel) and ASTM (Abstract Syntax Tree Metamodel) [9][10].

The success of approaches such as ADM and MDA depend on the existence of CASE tools that make a significant impact on software processes such as forward engineering and reverse engineering processes. The Eclipse Modeling Framework (EMF) was created for facilitating system modeling and the automatic generation of Java code [11]. EMF started as an implementation of MOF resulting Ecore, the EMF metamodel comparable to EMOF. EMF has evolved starting from the experience of the Eclipse community to implement a variety of tools and to date is highly related to MDD [12]. In this context, the subproject Model to Model Transformation (MMT), hosts model-to-model transformation languages. Transformations are executed by transformation engines that are plugged into the Eclipse Modeling infrastructure. For instance, Atlas Transformation Language (ATL) is a model transformation language and toolkit that provides ways to produce a set of target models from a set of source models [13]. Another subproject is Acceleo, which is an implementation of the Model-to-Text (M2T) transformation standard of the OMG for EMF-based models [14]. Acceleo is used in forward engineering processes.

Today, the most complete technology that support ADM is MoDisco, which provides a generic and extensible framework to facilitate the development of tools to extract models from legacy systems and use them on use cases of modernization. As an Eclipse component, MoDisco can integrate with plugins or technologies available in the Eclipse environment [15].

In the Eclipse environment, ADM is integrated with Java language but it is weakly supported for other programming languages such as C++ [11]. In particular, C++ is one of the most commonly used programming language in science and engineering domains and numerous legacy software components written in C++ require to be modernized. EMF4CPP is the first step at providing a set of tools for MDD in C++ as an alternative to the Eclipse tools for Java [16]. It is a C++ implementation and type mapping for the EMF core, the Ecore metamodel. The main facilities provided by EMF4CPP are to generate C++ code from Ecore metamodels and to parse and serialize models and metamodels from and into XMI documents [17]. However, an implementation of a MOF-compliant C++ metamodel would be necessary for other MDD processes (e.g., reverse engineering or software modernization).

In this work, we propose a migration process from C/C++ software to different mobile platforms that integrates ADM standards with Haxe. On the one hand, the process follows model-driven principles: all artifacts involved in the process are viewed as models and the process is viewed as a sequence of model-to-model transformations. On the other hand, Haxe easily adapts the native behaviors of the different platforms targeted in development projects enabling extremely efficient cross-platform development, ultimately saving time and resources. It is worth mention that an Ecore metamodel and a model injector for the C++ language are provided.

The article includes a simple case study, the migration of a C++ application “the Set of Mandelbrot” that allow us to exemplify the different steps of the process. We believe that our approach provides advantages over processes based only on traditional ad-hoc migration techniques increasing productivity due to the automation introduced in the generation of the new software.

The article is organized as follows. Section II presents related work highlighting our contribution. Section III, Background, briefly describes OMG standards for modernization and the Haxe and C++ metamodels. In Section IV, we present the migration process from C++ to mobile platforms. Section V details the different stages of the migration process through of a simple case study. Finally, in Sections VI and VII we present a critical analysis of our approach and conclusions respectively.

II. Related Work

In this section, existing approaches for the development of mobile applications that are related in some way with our approach are described.

Reference [18] proposes a new software architecture with the objective of providing the same service as mobile Web service as well as mobile application. The authors report on the feasibility study that they conducted in order to evaluate whether to use model-driven software development for developing mobile applications. They argue that the architecture is flexible enough to support mobile Web services and mobile applications at the same time. They have developed a metamodel to describe mobile application and have shown how to generate mobile application from that model.

The project BAMOS and an architecture designed and implemented for the generic and flexible development of mobile applications are described in [19]. The architecture is based on the declarative description of the available services. The authors describe a model-driven approach for generating almost the complete source code of mobile services.

Reference [20] goes through mobile development process and architectural structures and their analysis with empirical mobile application development. The architecture and architecture role on the development has been studied in mobile application and multiplatform service development.

Various authors describe challenges of mobile software development, for example, in [21] authors highlight creating user interfaces for different kinds of mobile devices, providing reusable applications across multiple mobile platforms, designing context aware applications and handling their complexity and, specifying requirements uncertainty. Issues related to ensuring that the application provides sufficient performance while maximizing battery life are remarked in [22].

A proposal for supporting mobile application development by using models as inputs to an emulator is outlined at [23]. The authors describe an MDD-based emulator for using in the design of graphical interfaces and interactions. They propose transform functional behavior and requirement models with design restrictions into emulated applications.

Reference [24] describes a DSL (Domain Specific Language), named MobDSL, to generate applications for multiple mobile platforms. They perform the domain analysis on two cases in the Android and iPhone platforms. This analysis allows inferring the basic requirements of the language defined by MobDSL.

A reengineering process that integrates traditional reverse engineering techniques such as static and dynamic analysis with MDA is presented at [25]. The article describes a case study that shows how to move CRM (Customer Relationship Management) applications from desktop to mobile platforms. The proposal was validated in the open source application platform Eclipse, EMF, EMP, ATL and Android platform. Reference [26] describes a migration process from Java to mobile platforms through the multiplatform language Haxe.

ANDRIU, a reverse engineering tool based on static analysis of source code for transforming user interface tiers from desktop application to Android, is described in [27]. ANDRIU has been developed for migrating traditional systems to Android applications although it was designed to be extended for different migrations to others mobile platforms.

Reference [28] describes six major trends affecting future smartphone design and use: personal computers, IoT, multimedia delivery, low power operation, wearable computing and context awareness.

Reference [29] describes a comprehensive tool suite called WebRatio Mobile Platform for model-driven development of mobile applications. It is based on an extended version of OMG standard language called IFML (Interaction Flow Modeling Language) empowered with primitives tailored to mobile systems that enable specification of mobile specific behaviors.

Reference [30] brings out the findings of the experiments carried out to understand the impact of application characteristics, cloud and architecture and the android emulator used, on application performance when the application is augmented to cloud.

Reference [31] presents a solution for facilitating the migration of applications to the cloud, inferring the most suitable deployment model for the application and automatically deploying it in the available Cloud providers.

A. Our Contribution: The Migration Process

In this article, we describe an original model-driven migration process based on ADM standards (Figure 1). The process includes:

1. Recovering the generic abstract syntax tree (AST) model, instance of GASTM, from code: this step is different for each programing language.
2. Transforming the AST model to a target model in the Haxe platform through an intermediate transformation to obtain the KDM model. The advantage of this intermediate step is that, starting from the KDM model it is possible to obtain high-level models such as UML class diagrams, activity diagrams and use cases diagrams. These models could be refactored and be the starting point for generating code. This step is common for each source language.
3. Generating Haxe from the Haxe model. Then, Haxe allows compilation of programs to multiple target languages such as Javascript, Neko, C++ and Java and to all major mobile platforms.

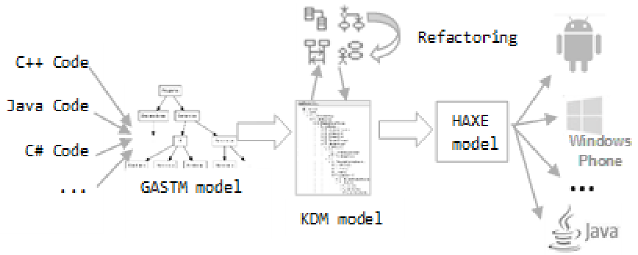


Fig. 1. Our contribution: The Migration Process

III. Background

In this section, we describe OMG standards for modernization. Next, we briefly describe the Haxe language and the Haxe metamodel. Finally, we describe the C++ metamodel that we defined via the Ecore metamodel.

A. Standards for Modernization

The purpose of standardization is to achieve well-defined interfaces and formats for interchange of information about software models to facilitate interoperability between the software modernization tools and services of the adherents of the standards. This will enable a new generation of solutions to benefit the whole industry and encourage collaboration among complementary vendors.

ADMTF is developing a set of standards of which we are interested in KDM and ASTM, both metamodels are defined via MOF. KDM is a metamodel for knowledge discovery in software that allows representing information related to existing software assets, their associations, and operational environments regardless of the implementation programming language and runtime platform. KDM is the foundation for software modernization representing entire enterprise software systems, not just code. ASTM is a specification for modeling elements to express abstract syntax trees. KDM and ASTM are two complementary modeling specifications. KDM establishes a specification that allows representing semantic information about a software system, whereas ASTM establishes a specification for representing the source code syntax by means of AST. ASTM acts as the lowest level foundation for modeling software within the OMG ecosystem of standards, whereas KDM serves as a gateway to the higher-level OMG models.

B. The Haxe Language

Haxe is an open-source high-level multiplatform programming language and compiler that can produce applications and source code for many different platforms from a single code-base [3].

Reference [32] summarizes the Haxe principles as follows: “support mainstream platforms”, “write once, reuse everywhere”, “always native, no wrapper”, “generated but readable” and “trust the developer”. Some languages allow cross-platform development, but neither their features nor their standard libraries are designed to run on multiple platforms. Haxe was designed from scratch to run and compile for many different platforms.

The Haxe programming language is a high level programming language that mixes features of object-oriented languages and functional ones. It is similar (but not pure) to object-oriented languages. The compiler supports novel features such as type inference, enforcing strict type safety at compile time.

Since language can be compiled for different platforms, it is useful for a wide variety of applications such as games, web and mobile. Haxe easily adapts the native behaviors of the different platforms targeted in development projects enabling extremely efficient cross-platform development, ultimately saving time and resources. Currently there are nine supported target languages: Javascript, Neko, PHP, Python, C++, Actionscript3, Flash, Java and, C#. In the context of Mobile App Development, Haxe allows writing mobile apps that target all major mobile platforms and run at native speed. The C++ target allows us to target Android or iOS, and OpenFL provides support for creating interfaces using a Flash-like API. OpenFL is a free and open source software framework and platform for the creation of multiplatform applications and video games [33]. OpenFL programs are written in Haxe and may be published to Flash movies, or standalone applications for Microsoft Windows, Mac OS X, Linux, iOS, Android, BlackBerry OS, Firefox OS, HTML5 and Tizen.

In previous work, we show an integration the Haxe with MDD defining an Ecore metamodel of the Cross-Platform Framework Haxe [26]. This metamodel allowed us to integrate Haxe with MDA migration process from Java or C/C++ to mobile platform.

C. The Haxe Metamodel

The Haxe metamodel conforms to Ecore and is partially shown in Figure 2. The main metaclasses of the Haxe metamodel are those that allow specifying an application using Haxe as language. One of the main metaclasses of the metamodel is *HaxeModel,* that serves as element container used to describe an application and store additional information on it, for example, some options of compilation and different metaclasses for modeling such as modules, classes and packages. *HaxeModel* owns *HaxeModule* and *HaxePathReferentiable*. Starting from the relations *haxeModules*, *referenced* and *elements*, the class *HaxeModel* allows storing different information. Relation *haxeModules* allows accessing the different Haxe modules used in the project. Through relation *elements*, it is possible to access the different elements of the package tree. Relation *referenced* provides access to elements, which are referenced in the project but are not defined completely. In the case of relations and referenced elements, the type used is *HaxePathReferentiable*, which is the parent type of metaclasses such as *HaxeType* and *HaxePackage*. The Haxe language includes different kind of types such as class (the types class and interface), function, abstract type, enumeration, and anonymous structures. The full Haxe metamodel can be found in [34].

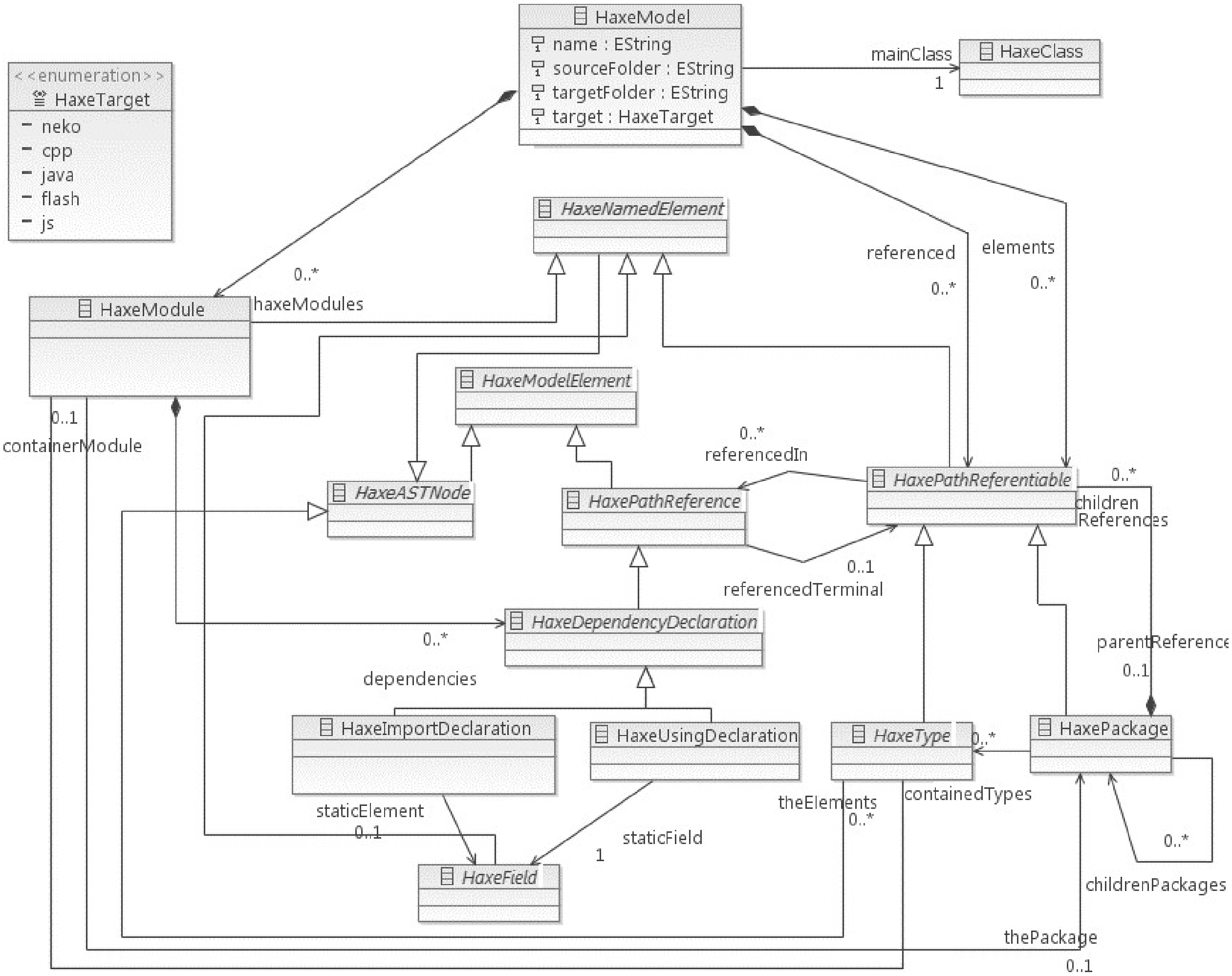


Fig. 2. The Haxe Metamodel

D. The C++ Metamodel

The C++ metamodel conforms to Ecore and is partially shown in Figure 3. The root metaclass is *Program* that represents a C++ program, which owns source files, instances of *TranslationUnit*. A translation unit contains declarations such as block declaration, function definitions and template declarations. A *SimpleDeclaration*, instance of *BlockDeclaration*, has a *DeclSpecifierSeq* that is a sequence of *DeclSpecifiers* which refers to a declaration specifiers and a type specifier. In addition, a simple declaration has an *InitDeclaratorList* containing a variable declaration list that is a list of specifiers and the name of a variable and its corresponding initialization. A *FunctionDefinition* has a *Declarator* containing the function identifier and the parameter list. *Function* and *CtorOrDestFunction*, instances of *FunctionDefinition*, have a body that contains a compound statement that owns statements such as declarations, iterations and selections. In addition, a *Function* has a *DeclSpecifierSeq* that is a sequence of *DeclSpecifiers* such as function specifiers and a type specifier. *TypeSpecifier* subclasses are *SimpleTypeSpecifier*, *ClassSpecifier* and *EnumSpecifier* among others. A *ClassSpecifier* has a *ClassHead* containing the class key (class or struct) and a *MemberSpecification* that contains *MemberDeclarations* such as variables, function declarations, function definitions, constructors, destructor, template members, etc.

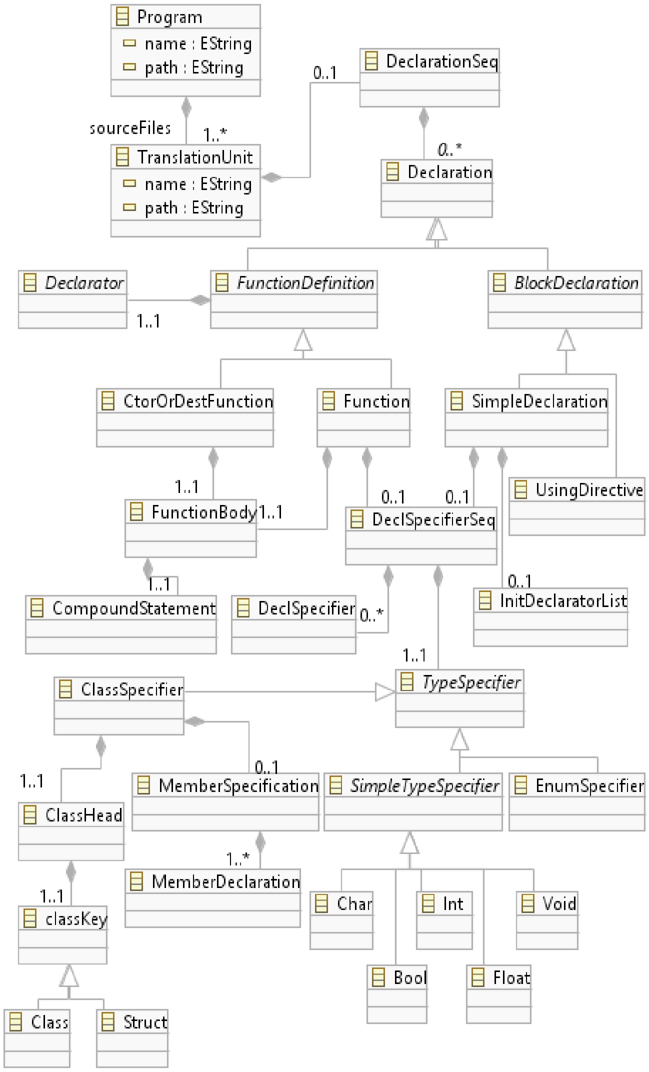


Fig 3. The C++ Metamodel.

IV. Migrating Legacy Code In The Adm Context

In this article, a process to migrate legacy code to Haxe in the ADM context is proposed. The migration process follows model-driven development principles: all artifacts involved in the process can be viewed as models that conforms to Ecore meta-metamodel, the process itself can be viewed as a sequence of model-to-model transformations and the extracted information is represented in a standard way through Ecore metamodels. For each transformation, source and target metamodels are specified. A source metamodel defines the family of source models to which transformations can be applied. A target metamodel characterizes the generated models. Figure 4 summarizes the proposed process.

The first step is the reverse engineering of source code to obtain the abstract syntax tree of the code and consists of two stages:

1. Generating the first model of the code by using a model injector. This model conforms to source code metamodel, such as C++ and Java. The obtained model could be refactored to reorganize and modify the syntactic elements to improve the design. The refactoring is implemented as a model-to-model transformation whose source and target models are instances of source code metamodel.
2. Generating the abstract syntax tree model, instance of the GASTM metamodel, from the model obtained in the previous stage by an ATL model-to-model transformation.

In this first step of the process, an injector and a transformation to obtain the GASTM model must be implemented for each language, whereas the sequence of transformations involved in the followings steps of the migration process are independent of the language of the legacy code.

The second step generates the KDM model. This process is carried out by means of an ATL model-to-model transformation that takes as input a model conforming to the GASTM metamodel and produces a model conforming to the KDM metamodel.

The next step is related to an ATL model-to-model transformation that generates a model of the Haxe platform from a KDM model. Then, it is possible to generate Haxe code from the Haxe model by using model-to-text transformations expressed in Acceleo. Considering that Haxe has one cross-platform standard library and various platform specific APIs used to natively access platform features, it is possible to write a mobile application once and have this application instantly available to different mobile devices.

The proposal was validated in the open source application platform Eclipse considering that some of its tools and run-time environments are aligned with ADM. Eclipse provides implementations of several metamodels such as Java, GASTM and KDM conforming to Ecore metamodel. We contribute with the implementation of C++ and Haxe metamodels, instances of Ecore metamodel. Model-to-model transformations were implemented in ATL that is a model transformation language in the field of MDE developed on top of the Eclipse platform. ATL is a hybrid language that provides a mix of declarative and imperative constructs.

V. Case Study: Migrating C++ Code To Mobile Platforms

In this section, we describe a migration process from C++ code to different mobile platforms through Haxe. This process starts extracting models from the C++ code. Next, these models are transformed into Haxe models that allow generating Haxe code which can be compiled to multiple target languages in a straightforward way.

To illustrate the migration process, we describe a simple case study, how to migrate the C++ code of “the Set of Mandelbrot” to Haxe code.

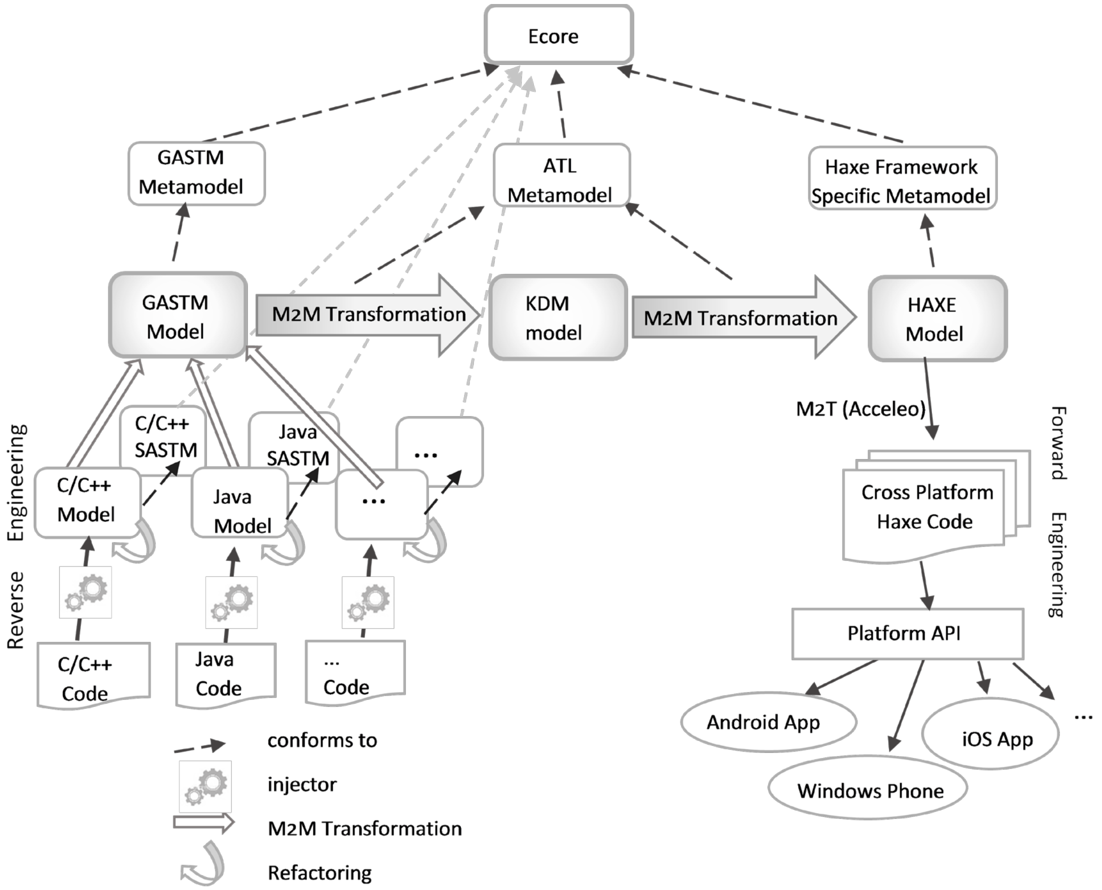


Fig. 4. The Migration Process.

The original application consists of a main class, called Mandelbrot, that is responsible for the calculation of the set of Mandelbrot and displaying it as image. To perform these tasks, the Mandelbrot class depends on Picture and Complex classes, the first is used as a data type that supports the manipulation of digital images. The second class is a data type used to model complex number with their respective operations. The following subsections describes the steps of the migration process.

A. Obtaining GASTM models from C++ code

Below, we describe the model-to-model transformations to generate generic abstract syntax tree (AST) models from C++ models.

This first transformation extracts an AST model specific to C++ from code. To carry out this task, we constructed a model injector by using EMFText [35]. To generate this injector, EMFText requires the language metamodel and the concrete syntax specification. In our approach, to generate the injector we first specified the C++ metamodel based on the C++ grammar [36]. Then, we specified the concrete syntax that defines the textual representation of all metamodel concepts. Taking these specifications, the EMFText generator derives an advanced textual editor that uses a parser and printer to parse language expressions to EMF models or to print EMF models to languages expressions respectively.

Figure 5 exemplifies the first step of the process. It partially shows C++ code of Mandelbrot Set, that is the input of the model injector, and the C++ model of the application in XMI format.

The second transformation takes as input the model obtained in the previous step and release a generic AST model conforming to the GASTM metamodel. This transformation specifies the way to produce GASTM projects (target) from C++ programs (source). Figure 6 partially shown the obtained GASTM model from the transformation. A project owns files that are obtained from the source files of the program. Each file, instance of *CompilationUnit*, owns fragments such as aggregate type definition, function definition and variable definition obtained from the translation of classes, function definitions, variables, etc.

B. Obtaining Haxe models

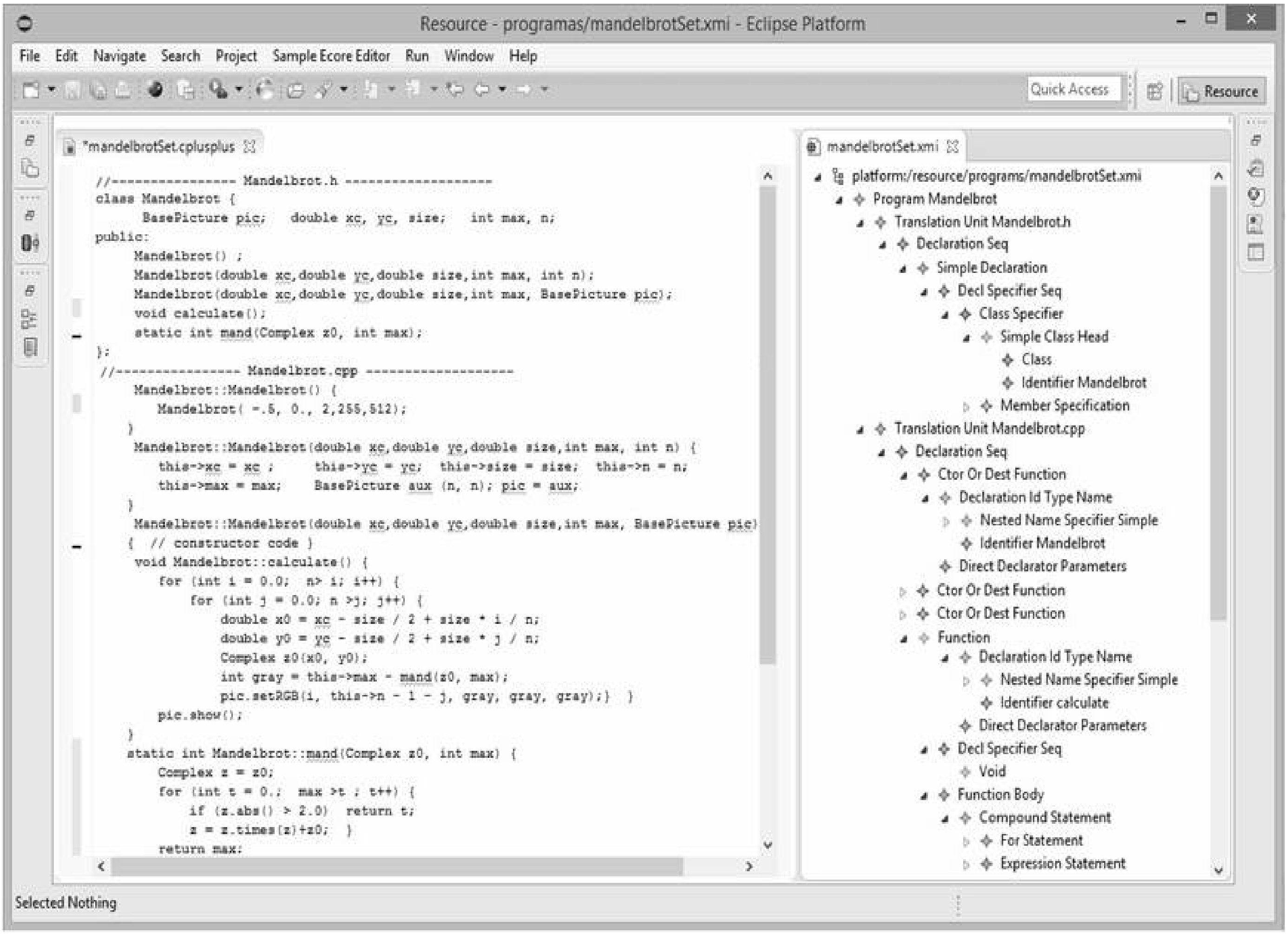
The previous transformations are dependent of the legacy code language, that is, for each language, the model injector and the transformation to obtain the generic AST model must be implemented.

In contrast to the previous stage, the sequence of transformations from GASTM models to Haxe models are independent of the language, that is, these transformations are common for all language of the legacy code:

- Transforming a GASTM model to a KDM model.
- Transforming a KDM model to a Haxe model: This transformation takes into account the characteristics of the Haxe language, for example, Haxe does not support neither multiple inheritance nor multiple class constructors.

These transformations are implemented in ATL and specified by means of ATL modules composed of the following elements:

- A header section that defines the names of the transformation module and the variables of the source and target metamodels.
- An optional import section that enables to import some existing ATL libraries.
- A set of helpers that can be used to define variables and functions.
- A set of rules that defines how source model elements are matched and navigated to create and initialize the elements of the target models.


Resource - programas/mandelbrotSet.xmi - Eclipse Platform
File Edit Navigate Search Project Sample Ecore Editor Run Window Help
Quick Access
Resource
*mandelbrotSet.cplusplus
mandelbrotSet.xmi
Selected Nothing

```
//---------------- Mandelbrot.h ------------------
class Mandelbrot {
      BasePicture pic;   double xc, yc, size;   int max, n;
public:
     Mandelbrot() ;
     Mandelbrot(double xc,double yc,double size,int max, int n);
     Mandelbrot(double xc,double yc,double size,int max, BasePicture pic);
     void calculate();
     static int mand(Complex z0, int max);
};
 //---------------- Mandelbrot.cpp ------------------
     Mandelbrot::Mandelbrot() {
        Mandelbrot( -.5, 0., 2,255,512);
     }
     Mandelbrot::Mandelbrot(double xc,double yc,double size,int max, int n) {
        this->xc = xc ;     this->yc = yc;  this->size = size;  this->n = n;
        this->max = max;    BasePicture aux (n, n); pic = aux;
     }
     Mandelbrot::Mandelbrot(double xc,double yc,double size,int max, BasePicture pic)
     {  // constructor code }
     void Mandelbrot::calculate() {
        for (int i = 0.0;  n> i; i++) {
            for (int j = 0.0; n >j; j++) {
                double x0 = xc - size / 2 + size * i / n;
                double y0 = yc - size / 2 + size * j / n;
                Complex z0(x0, y0);
                int gray = this->max - mand(z0, max);
                pic.setRGB(i, this->n - 1 - j, gray, gray, gray);}  }
        pic.show();
     }
    static int Mandelbrot::mand(Complex z0, int max) {
        Complex z = z0;
        for (int t = 0.;  max >t ; t++) {
            if (z.abs() > 2.0)  return t;
            z = z.times(z)+z0;  }
        return max;
```

```
platform:/resource/programs/mandelbrotSet.xmi
  Program Mandelbrot
    Translation Unit Mandelbrot.h
      Declaration Seq
        Simple Declaration
          Decl Specifier Seq
            Class Specifier
              Simple Class Head
                Class
                Identifier Mandelbrot
              Member Specification
    Translation Unit Mandelbrot.cpp
      Declaration Seq
        Ctor Or Dest Function
          Declaration Id Type Name
            Nested Name Specifier Simple
            Identifier Mandelbrot
          Direct Declarator Parameters
        Ctor Or Dest Function
        Ctor Or Dest Function
        Function
          Declaration Id Type Name
            Nested Name Specifier Simple
            Identifier calculate
          Direct Declarator Parameters
          Decl Specifier Seq
            Void
          Function Body
            Compound Statement
              For Statement
              Expression Statement
```

Fig. 5. The Mandelbrot Class: Code and Model.

To exemplify the ATL transformations, the GASTM-to-KDM transformation is partially shown in Figure 7. The module *ASTM2KDM* specifies the way to produce KDM models (target) from GASTM models (source). Some rules that carry out the transformation are the followings:

- The rule *Project2Segment* transforms each ASTM project into a KDM segment that owns models such as CodeModel containing code elements (data types, methods, variables, etc.) and InventoryModel that contains physical artifacts of the existing software system (source file, binary file, etc).
- The rule *AggregateTypeDefinition2ClassUnit* transforms each ClassType into a ClassUnit. Code elements are obtained from the variables and function of the original class; source is obtained from the source code location.
- The *VariableDefinition2StorableUnit* transforms each variable definition into a storable unit.
- The *FunctionDeclaration2MethodUnit* transforms each function declaration into a method Unit.

The resulting model of this transformation is partially shown in Figure 8.

C. Generating Haxe Code

From a model Haxe, it is possible to generate a source code in Haxe by using Acceleo. Haxe allows writing mobile applications that target all major mobile platforms in a straightforward way. The generated code is syntactically correct, although, it does not compile on other platforms without doing changes due to the code refers to proprietary technologies of C++. To run on mobile environments, these technologies can be replaced with OpenFL and HaxeUI (that is an open source, multi-platform application-centric user interface framework designed for Haxe and OpenFL). The code obtained is partially shown in Figure 9.

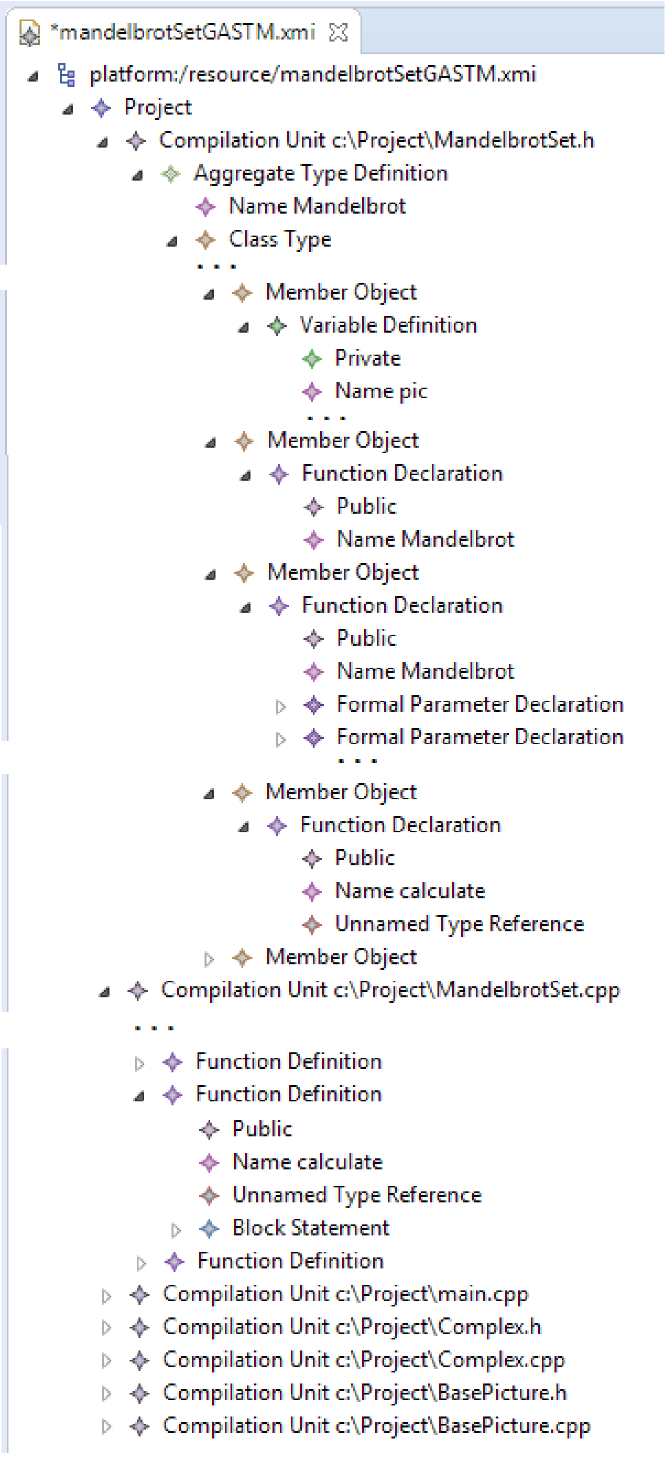
VI. Critical Analysis of Our Approach

This section analyzes critically our approach. First, we discuss advantages and limitations with regard to the application of an ADM approach. Next, we analyze the proposed migration process.

With regard to ADM, one of the well-known benefits is the increment of development productivity due to automation introduced in the generation of artifacts of the final system.

When a migration process is defined, it is important to consider that it is independent of the source and target technologies. In ADM, the intermediate models act as decoupling elements between source and target technologies. The independence is achieved with injectors and, M2M and M2T transformations. Besides, in a transformation sequence, models are an extension point to incorporate new stages.

ADM is based on MOF-like metamodeling that is a powerful approach for interoperability. For instance, a reverse engineering process recovers knowledge that must be represented using any formalism. For example, the XML technology seeks to solve the problem of expressing structured data in an abstract and reusable

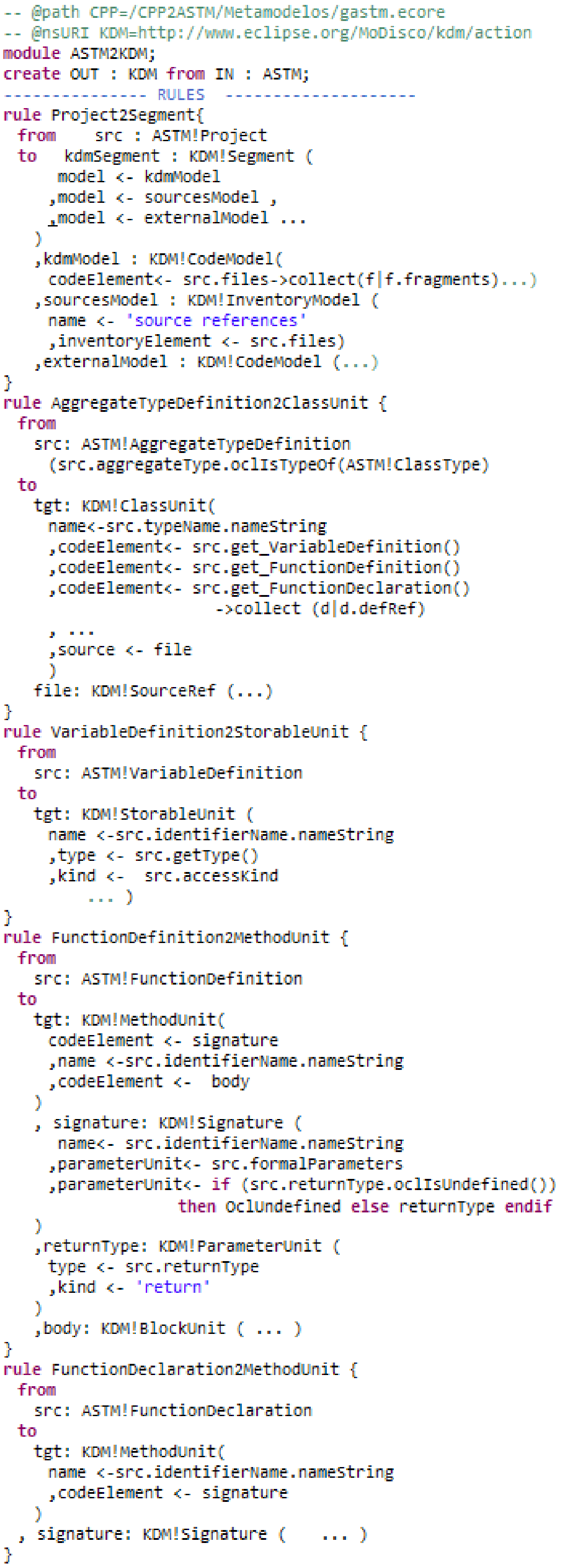


```
*mandelbrotSetGASTM.xmi
platform:/resource/mandelbrotSetGASTM.xmi
  Project
    Compilation Unit c:\Project\MandelbrotSet.h
      Aggregate Type Definition
        Name Mandelbrot
        Class Type
          . . .
          Member Object
            Variable Definition
              Private
              Name pic
              . . .
          Member Object
            Function Declaration
              Public
              Name Mandelbrot
          Member Object
            Function Declaration
              Public
              Name Mandelbrot
              Formal Parameter Declaration
              Formal Parameter Declaration
              . . .
          Member Object
            Function Declaration
              Public
              Name calculate
              Unnamed Type Reference
          Member Object
    Compilation Unit c:\Project\MandelbrotSet.cpp
      . . .
      Function Definition
      Function Definition
        Public
        Name calculate
        Unnamed Type Reference
        Block Statement
      Function Definition
    Compilation Unit c:\Project\main.cpp
    Compilation Unit c:\Project\Complex.h
    Compilation Unit c:\Project\Complex.cpp
    Compilation Unit c:\Project\BasePicture.h
    Compilation Unit c:\Project\BasePicture.cpp
```

Fig. 6. The Mandelbrot Set: GASTM Model.

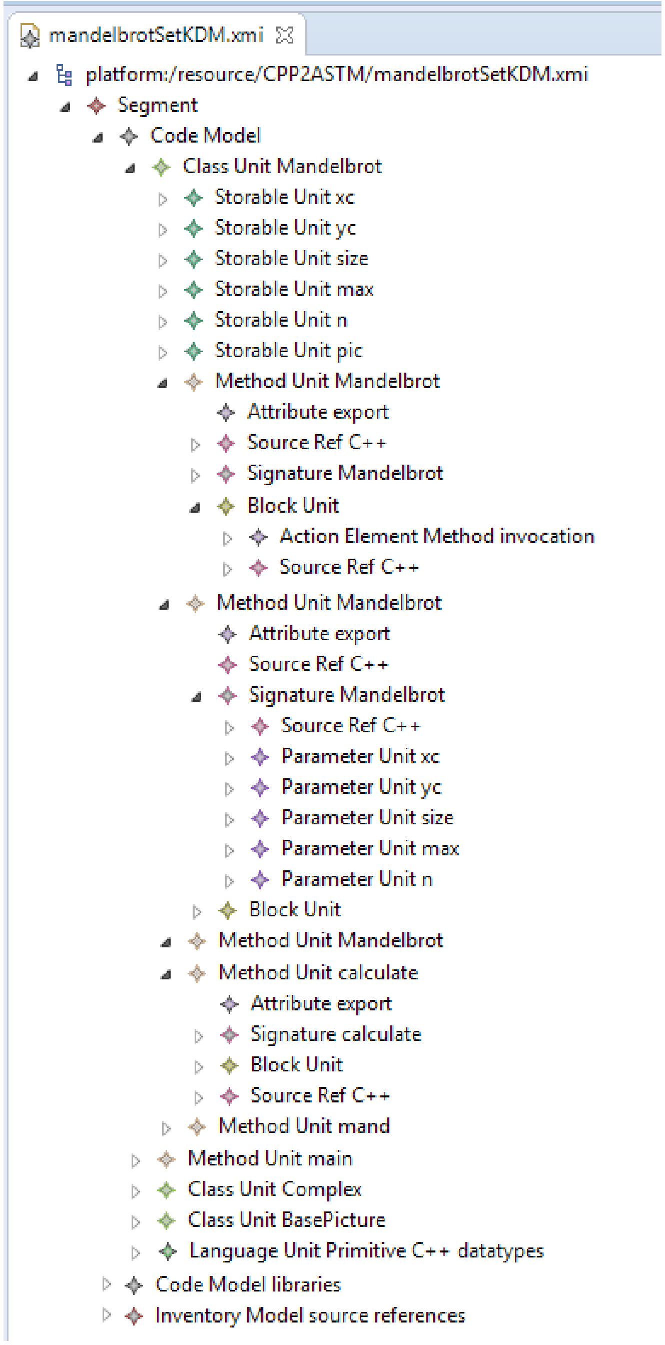
way. Metamodeling languages as MOF or its implementation called Ecore will outperform XML for its expressive power and the existence of powerful model transformation languages to implement transformations that are required at the different stages of the migration process. MOF-like metamodeling also includes the possibility to attach OCL restrictions to complete the model specification. In addition, MOF-like metamodels allow a clear separation of abstract and concrete syntax and thus, associate different notations for a model.

On the other hand, we can mention the following limitations of ADM. There are no available open and free injectors for different languages and it is often necessary to implement them. As regard



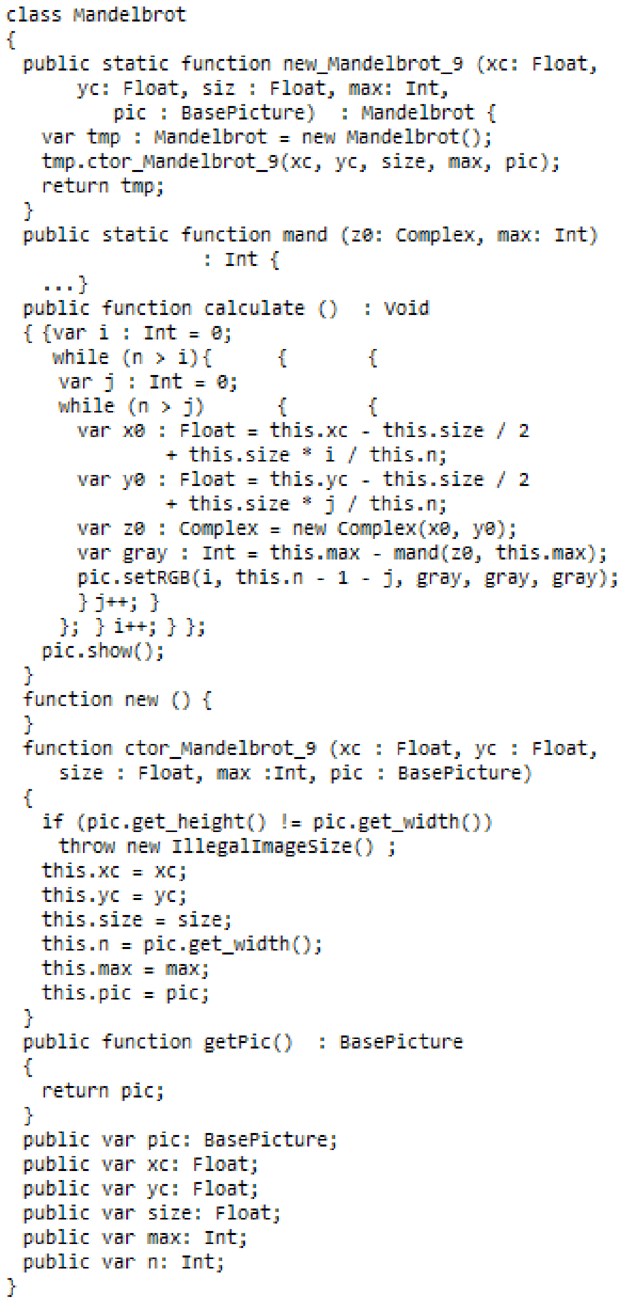
```
-- @path CPP=/CPP2ASTM/Metamodelos/gastm.ecore
-- @nsURI KDM=http://www.eclipse.org/MoDisco/kdm/action
module ASTM2KDM;
create OUT : KDM from IN : ASTM;
--------------- RULES  --------------------
rule Project2Segment{
 from    src : ASTM!Project
 to   kdmSegment : KDM!Segment (
     model <- kdmModel
    ,model <- sourcesModel ,
    ,model <- externalModel ...
   )
   ,kdmModel : KDM!CodeModel(
    codeElement<- src.files->collect(f|f.fragments)...)
   ,sourcesModel : KDM!InventoryModel (
    name <- 'source references'
    ,inventoryElement <- src.files)
   ,externalModel : KDM!CodeModel (...)
}
rule AggregateTypeDefinition2ClassUnit {
 from
   src: ASTM!AggregateTypeDefinition
    (src.aggregateType.oclIsTypeOf(ASTM!ClassType)
 to
   tgt: KDM!ClassUnit(
    name<-src.typeName.nameString
    ,codeElement<- src.get_VariableDefinition()
    ,codeElement<- src.get_FunctionDefinition()
    ,codeElement<- src.get_FunctionDeclaration()
                     ->collect (d|d.defRef)
    , ...
    ,source <- file
    )
   file: KDM!SourceRef (...)
}
rule VariableDefinition2StorableUnit {
 from
   src: ASTM!VariableDefinition
 to
   tgt: KDM!StorableUnit (
    name <-src.identifierName.nameString
    ,type <- src.getType()
    ,kind <-  src.accessKind
        ... )
}
rule FunctionDefinition2MethodUnit {
 from
   src: ASTM!FunctionDefinition
 to
   tgt: KDM!MethodUnit(
    codeElement <- signature
    ,name <-src.identifierName.nameString
    ,codeElement <-  body
   )
   , signature: KDM!Signature (
     name<- src.identifierName.nameString
    ,parameterUnit<- src.formalParameters
    ,parameterUnit<- if (src.returnType.oclIsUndefined())
                   then OclUndefined else returnType endif
   )
   ,returnType: KDM!ParameterUnit (
    type <- src.returnType
    ,kind <- 'return'
   )
   ,body: KDM!BlockUnit ( ... )
}
rule FunctionDeclaration2MethodUnit {
 from
   src: ASTM!FunctionDeclaration
 to
   tgt: KDM!MethodUnit(
    name <-src.identifierName.nameString
    ,codeElement <- signature
   )
  , signature: KDM!Signature (   ... )
}
```

Fig 7. The ASTM2KDM Transformation.



```
mandelbrotSetKDM.xmi
platform:/resource/CPP2ASTM/mandelbrotSetKDM.xmi
  Segment
    Code Model
      Class Unit Mandelbrot
        Storable Unit xc
        Storable Unit yc
        Storable Unit size
        Storable Unit max
        Storable Unit n
        Storable Unit pic
        Method Unit Mandelbrot
          Attribute export
          Source Ref C++
          Signature Mandelbrot
          Block Unit
            Action Element Method invocation
            Source Ref C++
        Method Unit Mandelbrot
          Attribute export
          Source Ref C++
          Signature Mandelbrot
            Source Ref C++
            Parameter Unit xc
            Parameter Unit yc
            Parameter Unit size
            Parameter Unit max
            Parameter Unit n
          Block Unit
        Method Unit Mandelbrot
        Method Unit calculate
          Attribute export
          Signature calculate
          Block Unit
          Source Ref C++
        Method Unit mand
      Method Unit main
      Class Unit Complex
      Class Unit BasePicture
      Language Unit Primitive C++ datatypes
    Code Model libraries
    Inventory Model source references
```

Fig 8. The Mandelbrot Set: The KDM Model.



```
class Mandelbrot
{
  public static function new_Mandelbrot_9 (xc: Float,
        yc: Float, siz : Float, max: Int,
            pic : BasePicture)  : Mandelbrot {
    var tmp : Mandelbrot = new Mandelbrot();
    tmp.ctor_Mandelbrot_9(xc, yc, size, max, pic);
    return tmp;
  }
  public static function mand (z0: Complex, max: Int)
                    : Int {
    ...}
  public function calculate ()  : Void
  { {var i : Int = 0;
     while (n > i){      {        {
      var j : Int = 0;
      while (n > j)      {        {
        var x0 : Float = this.xc - this.size / 2
                 + this.size * i / this.n;
        var y0 : Float = this.yc - this.size / 2
                 + this.size * j / this.n;
        var z0 : Complex = new Complex(x0, y0);
        var gray : Int = this.max - mand(z0, this.max);
        pic.setRGB(i, this.n - 1 - j, gray, gray, gray);
        } j++; }
      }; } i++; } };
    pic.show();
  }
  function new () {
  }
  function ctor_Mandelbrot_9 (xc : Float, yc : Float,
      size : Float, max :Int, pic : BasePicture)
  {
    if (pic.get_height() != pic.get_width())
      throw new IllegalImageSize() ;
    this.xc = xc;
    this.yc = yc;
    this.size = size;
    this.n = pic.get_width();
    this.max = max;
    this.pic = pic;
  }
  public function getPic()  : BasePicture
  {
    return pic;
  }
  public var pic: BasePicture;
  public var xc: Float;
  public var yc: Float;
  public var size: Float;
  public var max: Int;
  public var n: Int;
}
```

Fig 9. The Mandelbrot Set: Haxe Code.

KDM metamodel, it is designed to support interoperability between modernization tools. For some aspects such as user interfaces, KDM provides a reduced level of detail and does not allow to express common concepts to many technologies. This requires to extend the metamodel (which leaves to conform to a standard) or define stereotypes. With regard to model transformation languages, they suffer limitations due to it does not allow defining complex data structures and algorithms.-

Regarding the proposed migration process, we show the viability of semi-automatic migration processes based on ADM. Due the fact that the objective of the migration is not only "compile" an application in a mobile platform but also to create a modified version of the application using quality criteria, the process can not be fully automated. Next, we informally compare the model-driven migration process with brute-force redevelopment migration.

Our approach involves preliminary activities that require time and cost, for instance we need to define metamodels and transformations if they do not exist. It is assumed that using a brute-force redevelopment, developers do not need training to write metamodels and model transformations. However, model transformations allow developers to concentrate on conceptual aspects of the relations between models and then to delegate most of the migration process to the transformation rules, whereas in the brute-force redevelopment, developers need to migrate by hand the legacy systems, making over and over again the same task. It is worth mentioning that the generation of models by ATL transformations, aims to generate models "Correct-by-Construction" with regard to metamodel specifications.

A general limitation on migration processes is the cost of testing due to these activities in general are handled manually. In the context of model-driven approaches, the cost of testing could be reduced by defining semiautomatic process based on metamodels.

Beyond the previous issues, we consider that mobile application developers need frequently adapt software components and applications developed in languages such as Java or C/C++. Then, model-driven migration processes could be reused and the cost of preliminary activities is recovered.

VII. Conclusion

This paper describes an approach for adapting object-oriented software in C/C++ to mobile platforms. A migration process, based on the integration of ADM and the HAXE platform, has been proposed. The main contributions of our approach is a sequence of transformations implemented to migrate C++ code to mobile platform based on ADM standards allowing reusing both the transformations and the generated models. Besides, we provide a definition of the C++ metamodel via the Ecore metamodel and the implementation of an injector to obtain the first model from C++ code.

We believe that our approach provides benefits with regard to processes based only on traditional migration techniques. The migration process can be divided in smaller steps focusing in specific activities, and be automated thanks to the chaining of model transformations. All the involved artifacts can be reused, modified for evolution purposes or extended for other purposes. The metamodel approach enables covering different levels of abstraction and satisfying several degrees of detail depending on the needs of the migration and is the key for interoperability. All artifacts can be actually represented as models so that there is no information loss during the migration process. Model transformations allow developers to concentrate on the conceptual aspects of the relations between models and delegate the implementation of the transformation.

The proposal was validated in the open source application platform Eclipse considering that some of its tools and run-time environments are aligned with ADM standards. Our approach has already shown to work on real applications of medium size. We foresee to apply our approach in real industrial projects.

References

[1] Gartner, http://www.gartner.com/it-glossary/internet-of-things/, 2016.

[2] M. Weiser. The Computer for the 21st Century, Scientific American, Vol. 265 No.9, pp. 66-75, 1991.

[3] B. Dasnois. *Haxe 2 Beginner's Guide*. Packt Publishing, 2011.

[4] M. Brambilla, J. Cabot, M. Wimmer. *Model-Driven Software Enginneering in Practice*, Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, 2012.

[5] MDA. The Model-Driven Architecture. http://www.omg.org/mda/, 2016.

[6] *Meta Object Facility (MOF) Core Specification*, Version 2.5, OMG Document Number: formal/2015-06-05. Available: http://www.omg.org/spec/MOF/2.5/

[7] *OMG Object constraint language* (OCL), version 2.4. OMG Document Number: formal/2014-02-03. Available: http://www.omg.org/spec/OCL/2.4

[8] ADM. Architecture-driven modernization task force. http://www.adm.org, 2016.

[9] *Knowledge Discovery Meta-Model* (KDM), OMG Document Number: formal/2011-08-04. Available: http://www.omg.org/spec/KDM/1.3, 2011.

[10] *Abstract Syntax Tree Metamodel*, version 1.0, OMG Document Number: formal/2011-01-05. Available: http://www.omg.org/spec/ASTM, 2011.

[11] D. Steinberg, F. Budinsky, M. Paternostro, E.Merks. *EMF: Eclipse Modeling Framework* (2nd ed.). Addison-Wesley, 2009.

[12] EMF. Eclipse Modeling Framework (EMF). http://www.eclipse.org/modeling/emf/ 2016.

[13] ATL. Atlas Transformation Language Documentation. http://www.eclipse.org/atl/documentation/, 2016.

[14] Acceleo. Obeo. Acceleo Generator. http://www.eclipse.org/Acceleo/, 2016.

[15] MoDisco. https://eclipse.org/MoDisco/, 2016.

[16] EMF4CPP: What is EMF4CPP? https://code.google.com/archive/p/emf4cpp/, 2016.

[17] *XML Metadata Interchange (XMI) Specification*, OMG Document Number: formal/2015-06-07. Available: http://www.omg.org/spec/XMI/2.5.1

[18] P. Braun, R. Eckhaus, "Experiences on model-driven software development for mobile applications" in *Proc. of Engineering of Computer-Based Systems, IEEE International Conference and Workshop on the Engineering of computer Base Systems ,* Washington, DC, USA, IEEE Computer Society, 2008, pp. 490-493.

[19] J. Dunkel, R. Bruns, "Model-driven architecture for mobile applications". *Business Information Systems* (LNCS), Berlin: Springer-Verlag, 2007, vol. 4439, pp. 464-477.

[20] H. K. Kim, Frameworks of process improvement for mobile applications. *Engineering Letters*, 16(4), 2008, 550-555. Available: http://www.engineeringletters.com/issues_v16/issue_4/EL_16_4_13.pdf

[21] J. Dehlinger, J. Dixon. "Mobile application software engineering: Challenges and research directions" in *Proc. of the Workshop on Mobile Software Engineering,* Berlin, Springer-Verlag, 2011, pp. 29-32.

[22] C. Thompson, D. Schmidt, H. Turner,J. White, "Analyzing Mobile Application Software Power Consumption via Model-Driven Engineering", *Advances and Applications in Model-Driven Engineering,* Chapter 16, IGI GLOBAL, 2014, pp.342-366.

[23] J. Bowen, A. Hinze, "Supporting mobile application development with model-driven emulation", *Journal of the ECEASST*, vol. 45, 2011, pp. 1–5.

[24] D. Kramer, T. Clark, S. Oussena, "MobDSL: A domain specific language for multiple mobile platform deployment", in *IEEE Int. Conf. on Networked Embedded Systems for Enterprise Applications* (NESEA), 2010, pp. 1-7.

[25] F. Améndola, L. Favre, "Adapting CRM systems for mobile platforms: An MDA perspective", *14th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (SNPD´13), 2013, pp. 323-328.

[26] P. Diaz Bilotto, L. Favre, "Migrating Java to Mobile Platforms through Haxe: An MDD Approach", Chapter 13, *Modern Software Engineering Methodologies for Mobile and Cloud Environments*. Antonio Miguel Rosado da Cruz, Sara Paiva, eds. (pp.240-268), IGI GLOBAL, 2016.

[27] R. Pérez Castillo, I. García Rodriguez, R. Gómez Cornejo, M. R. Pérez Castillo, I. García Rodriguez, R. Gómez Cornejo, M. Fernández Ropero, M. Piattini, ANDRIU. A Technique for Migrating Graphical User Interfaces to Android. In *Proc. of The 25th International Conference on Software Engineering and Knowledge Engineering* (SEKE 2013), Boston: Knowledge Systems Institute, pp. 516-519.

[28] N. Islam, R. Want. Smarthphones: Past, present and future. *Pervasive Computing*, 13(4), 2014, pp.82-92.

[29] R. Acerbis, A. Bongio, M. Brambilla, S. Butti, Model-Driven Development Based on OMG's IFML with WebRatio Web and Mobile Platform. *Engineering the Web in the Big Data Era*. Lecture Notes in Computer Science, vol. 9114, 2015, pp. 605-608.

[30] P. Joshi, A. Nivangune, R. Kumar, S. Kumar, R. Ramesh, S. Pani, A. Chesum, Understanding the Challenges in Mobile Computation Offloading to Cloud through Experimentation. In *2nd ACM Int. Conf. on Mobile Software Engineering and Systems* (MOBILESoft), 2015, pp.158-159.

[31] J. Ejarque, A. Micsik, R. M. Badia, "Towards Automatic Application Migration to Clouds", in *IEEE 8th Int. Conf. on Cloud Computing* (CLOUD), 2015, pp. 25-32.

[32] N. Cannasse, Haxe. Too Good to be True? GameDuell Tech Talk. http://www.techtalk-berlin.de/news/read/nicolas-cannasse-introducing-Haxe/, 2014.

[33] OPEN FL, http://www.openfl.org/, 2016.

[34] P. Diaz Bilotto. "Software development for mobile applications through an integration of MDA and Haxe*"*. Undergraduate Thesis. Computer Science Department, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, 2015.

[35] EMFText, www.emftext.org , 2016.

[36] B. Stroustrup, *The C++ Programming Language*. Addison-Wesley, Third Edition, 1997.

Liliana Martinez has a Master degree in Software Engineering. She is an assistant professor in Computer Science area at the Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil, Argentina. She is a member of the Software Technology Group, which develops its activities at the INTIA Research Institute at the UNCPBA. Her research interests are focused on system modernization, reverse engineering in the ADM context in particular. She has published book chapters, journal articles and conference papers. She has been member of the program committee of international conferences related to software engineering.

Claudia Pereira is an assistant professor in Computer Science area at the Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil, Argentina. She is a member of the Software Technology Group, which develops its activities at the INTIA Research Institute at the UNCPBA. She has a Master degree in Software Engineering from Universidad Nacional de La Plata, Argentina. Her research interests are focused on system modernization. She has published book chapters, journal articles and conference papers. She has been member of the program committee of international conferences related to software engineering.

Liliana Favre is Liliana Favre is a full professor of Computer Science at Universidad Nacional del Centro de la Provincia de Buenos Aires in Argentina. She is also a researcher of CIC (Comisión de Investigaciones Científicas de la Provincia de Buenos Aires). Her current research interests are focused on model driven development, model driven architecture and formal approaches, mainly on the integration of algebraic techniques with MDA-based processes. She has been involved in several national research projects about formal methods and software engineering methodologies. Currently she is research leader of the Software Technology Group at Universidad Nacional del Centro de la Provincia de Buenos Aires. She has published several book chapters, journal articles and conference papers. She has acted as editor of the book UML and the Unified Process. She is the author of the book Model Driven Architecture for Reverse Engineering Technologies: Strategic Directions and System Evolution.

Taxonomies for Reasoning About Cyber-physical Attacks in IoT-based Manufacturing Systems

[1]Yao Pan, [1]Jules White, [1]Douglas C. Schmidt, [2]Ahmad Elhabashy, [2]Logan Sturm, [2]Jaime Camelio, and [2]Christopher Williams

[1]*Vanderbilt University, USA*
[2]*Virginia Tech, USA*

***Abstract* — The Internet of Things (IoT) has transformed many aspects of modern manufacturing, from design to production to quality inspection. In particular, IoT and digital manufacturing technol-ogies have substantially accelerated product development-cycles and manufacturers can now create products of a complexity and precision not heretofore possible. However, new threats to supply chain security have arisen from connecting machines to the In-ternet and introducing complex IoT-based systems controlling manufacturing processes. By attacking these IoT-based manufacturing systems and tampering with digital files, attackers can manipulate physical characteristics of parts and change the dimensions, shapes, or mechanical properties of the parts, which can result in parts that fail in the field. These defects increase manufacturing costs and allow silent problems to occur only under certain loads that can threaten safety and/or lives. To understand potential dangers and protect manufacturing system integrity, this paper presents two taxonomies: one for classifying cyber-physical attacks against manufacturing processes and another for quality inspection measures for counteracting these attacks. We systematically identify and classify possible cyber-physical attacks and connect the attacks with variations in manufacturing processes and quality inspection measures. Our taxonomies also provide a scheme for linking emerging IoT-based manufacturing system vulnerabilities to possible attacks and quality inspection measures.**

***Keywords* — Cyber-physical attack, Computer-aided Manufacturing, Cyber-physical system, Internet of Things.**

I. Introduction

The Internet of Things (IoT) embeds electronics, software, and sensors into physical objects that collect and exchange data via network connections. IoT technologies have made manufacturing smarter by enabling manufacturing systems to evolve from loose collections of largely disjoint cyber and physical components into synergistic cyber-physical systems. The Internet-connected sensors, tooling, and control systems forming these IoT-based manufacturing systems enable the manufacturing and refinement of parts that previously were hard to produce cost-effectively.

The IoT plays an important role in improving the efficiency and productivity of manufacturing systems. For example, by connecting digital manufacturing technologies and Computer-Aided Engineering (CAE) tools, designers and manufacturing engineers can substantially accelerate the product development-cycle. The use of IoT-based manufacturing systems, however, also expands opportunities for cyber-physical attacks against these systems. In particular, older pre-IoT equipment was not Internet-accessible and thus not exposed to cyber-attack like newer IoT-based manufacturing equipment.

For instance, with IoT-based manufacturing systems, critical manufacturing files are stored in computers connected to the Internet, as shown in Fig. 1. It is possible for an attacker across the Internet to remotely intercept and alter design files or machine configurations to create undetectable changes in a part that adversely affect a product's design intent, performance, or quality [1], [2], [3], [4]. Since the parts being attacked are installed in automobiles, jet engines, or artificial heart valves, the results could financially devastate manufacturers, *e.g.*, by damaging equipment, incurring property losses, increasing warranty costs, losing customer trust, or threating human safety if these altered parts function improperly and fail in the field [2].

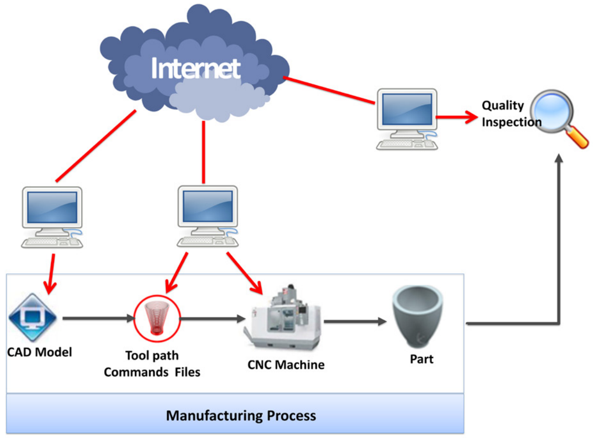


Fig. 1. Computers with Internet Connection in Manufacturing Systems.

A fundamental concern with IoT-based manufacturing systems is that they enable the monitoring and control of previously non-remotely accessible physical systems. If these Internet-connected IoT devices are not protected, the physical systems that they influence, such as the parts that a manufacturing facility produces, may be damaged. A famous example of critical IoT-based infrastructure being attacked is the Stuxnet malware that damaged nearly one-fifth of Iran's nuclear centrifuges [5]. The Stuxnet malware targeted programmable logic controllers and forced physical equipment to operate outside its design tolerances and led to centrifuge failures.

Past IoT security research has explored cyber-vulnerabilities in industrial control systems, such as Supervisory Control and Data Acquisition (SCADA) controllers [6], which can force physical systems to operate outside of their safety tolerances. While these control systems are a crucial area of research, IoT-based manufacturing systems are also vulnerable to silent attacks that result in a manufactured part's physical characteristics no longer matching their design specifications, which could lead to critical and/or pre-mature failures in the field. Similar research has looked at flaws injected into computer hardware and software logic [7], [8]. Much less research, however, has focused on flaws injected into the physical parts themselves, which have no computational logic.

DOI: 10.9781/ijimai.2017.437

In contrast to traditional cyber-security, IoT-based manufacturing systems use physical equipment, which generates measurable phenomena (e.g., temperatures and vibrations) to produce physical products that can be inspected and tested to determine if they meet their requirements. In addition, a particularly vexing challenge of IoT-based manufacturing systems is that their underlying software and hardware is rarely updated [1], [2], [3]. This lack of updates leaves complex IoT-based manufacturing equipment exposed and vulnerable to attack on the Internet. Moreover, this update problem cannot be easily addressed, as IoT-based manufacturing equipment is often extremely costly to purchase, amortized over decades, and very expensive to take out of production operation. Techniques and tools are therefore needed to help protect the physical parts that IoT-based manufacturing systems produce, while recognizing that these systems will always be at risk of cyber-attacks.

Fortunately, cyber-physical attacks against an IoT-based manufacturing system are unique in having correlated cyber *and* physical manifestation of the attack in the manufactured part. This correlation can be used to model and predict the relationships between attacks, process data, product quality observations, and side-channel impacts for the purpose of attack detection and diagnosis.

Hence, the work presented in this paper helps answer the following questions:

- What types of attacks are particular IoT-based manufacturing system processes vulnerable to?
- What facets of a part can be attacked in a given IoT-based manufacturing system?
- What quality inspection mechanisms could be put in place to lower risk in IoT-based manufacturing systems?
- How can quality inspection and side channel measurements mitigate cyber-vulnerabilities in IoT-based manufacturing system?
- How does a newly disclosed cyber vulnerability impact a particular IoT-based manufacturing process?

To answer these questions, we have created two taxonomies: one for classifying cyber-physical attacks against IoT-based manufacturing processes and another for quality inspection measures for counteracting these attacks. These taxonomies catalog IoT-based manufacturing processes, attacks, and quality inspection measures, as well as model the relationship between specific attack types, vulnerabilities, equipment, processes, and quality inspection measures. They also help to bridge the gap between (1) the IoT *cyber domain*, where the research subjects are cyber infrastructure and software vulnerabilities, and (2) the *physical domain*, which includes manufacturing processes and quality inspection measures.

Our taxonomies provide a framework that researchers and practitioners from both cyber-security and IoT-based manufacturing can use and augment to understand the scope of cyber vulnerabilities, how these vulnerabilities impact different processes, the types of cyber attributes that these attacks express, and their impacts on the physical properties of both the process execution and physical part outputs. This framework makes it easier to make decisions on cyber-physical security in manufacturing, catalog attacks and vulnerabilities as they emerge, and understand the relationship between specific attack types, equipment, processes, and side-channel impacts.

The remainder of this paper is organized as follows: Section II describes the taxonomies for the manufacturing process, cyber-physical attacks, and quality inspection measures; Section III explores a case study of a manufacturing industry partner using the proposed taxonomy; Section IV compares our research with related work; and Section V presents concluding remarks and future work.

II. Taxonomies

Our overarching goal is to connect vulnerabilities, IoT-based manufacturing processes, cyber-physical attacks and quality inspection measures all together, as we show in Fig. 2. The characteristics of the IoT-based manufacturing processes reveal the vulnerabilities that could be exposed, which would then determine what cyber-physical attacks could be launched. Each cyber-physical attack has its effects either in the physical domain or the cyber domain. We can choose the quality inspection measures that could capture the provisioned attack effects, thereby enabling better defenses against cyber-physical attacks in IoT-based manufacturing systems.

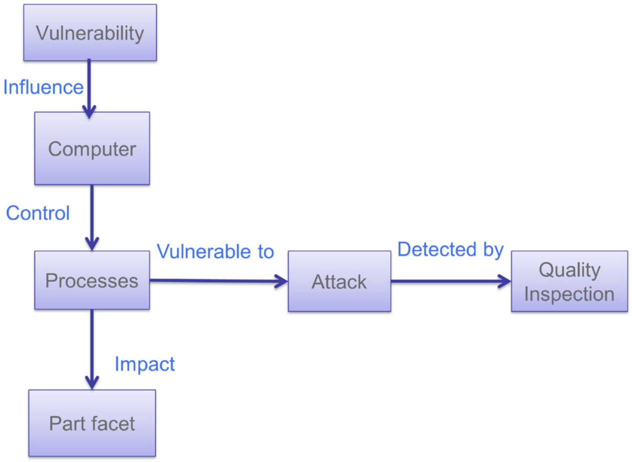


Fig. 2. Logic Flow in Manufacturing Processes.

A. Overview of Manufacturing Processes

Manufacturing systems are rarely the same for different types of manufactured products, but most of these systems share a similar workflow. A manufacturing system typically starts with product design, then procures raw material, goes through various manufacturing processes, followed by assembly and inspection for quality control, and finally distribution of the products, as shown in Fig. 3. Our taxonomies focus on the chain of process steps ranging from design to manufacturing with its different operations to inspection only, without considering other steps such as raw material procurement and distribution.

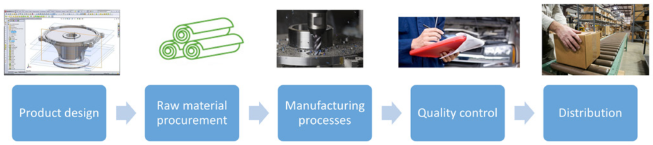


Fig. 3. Workflow of Manufacturing System.

A key differentiator between IoT-based manufacturing systems and traditional systems is that the former operate more like distributed software-reliant systems than the latter. Traditional manufacturing systems use significant numbers of manual steps and closed/locally managed control systems. Newer IoT-based manufacturing systems are remotely accessible and monitorable by designers, reconfigurable, and capture volumes of sensor and tool actuation data during operation. Moreover, these systems are driven by computer instructions that coordinate their constituent IoT sensors and tooling to produce a given part.

Since IoT-based manufacturing processes perform the set of steps through which raw materials are transformed into a finished product, this sub-section summarizes the basic and most commonly used manufacturing processes in industry today. In production systems, a combination of several processes may be required to manufacture a product, but understanding the characteristics of the essential and most common processes is important to build accurate taxonomies.

There are several methods [9], [10] [11] to classify the different manufacturing processes involved in production, such as dividing them into the two main groups shown in Fig. 4:

1. ***Processing operations***, which add value to materials by transforming them from one state to another. Process operations can be further divided into *solidification processes* (such as casting that pours material in a cavity to fill when it cools down), *deformation processes* (such as forming that changes the shape of the material, without usually changing its original volume), *subtractive processes* (such as machining that changes the shape of the material through removing some of it, thereby decreasing its volume), *additive processes* (such as 3D printing that builds the shape of the material progressively by accumulating thin layers one on top of the other), *surface processing* (such as surface finishing done as a final step to improve the quality of the surface of the current product), and others (such as heat treatment, which enhances the property of the material itself, and particulate processing, where particles are consolidated together).
2. ***Joining operations***, which bring two or more components together. Joining operations can be split into permanent joining processes (such as welding) and joining via mechanical components (such as fasteners).

An overview of such grouping can be seen in Fig. 4, along with some (non-exhaustive) examples for each sub-group. These sub-groups are not necessarily mutually exclusive, *e.g.*, a subtractive process may also be performed during surface processing operations.

Another concept we define is "part facet", which is a specific aspect or geometric structure of a part that is important to its performance. The facet type includes dimension (*e.g.*, length, width, height, radius, etc.), weight, center of gravity, color, magnetism, surface roughness, tensile strength, yield strength, etc. Each manufacturing process is restricted by its characteristics, so it can only affect a subset of the part facets. For example, a turning process can change the dimensions of the part. Likewise, a heat treatment can change the yield strength of the part.

B. Design Artifacts to Code

An interesting facet of IoT-based manufacturing is that design files, such as solid geometry representations of parts, are eventually translated into computer instructions, such as G-Code, for a set of IoT machines indicating how to manufacture the part [12]. This process is a form of model-driven engineering, which is also used in software development [13]. Many of the attacks are analyzed based on the instruction set limitations of manufacturing equipment, which are directly connected to the physical capabilities of the equipment, and provide cyber-physical bounds on attacks.

Due to the wide range of IoT-based manufacturing processes, this paper only concentrates on subtractive and additive processes, which serve as representatives of a larger group due to the fact that they are currently being used heavily in IoT environments. For example, in Computer Aided Manufacturing (CAM) the products within these processes are created through Computer Aided Design (CAD) software. The design is then realized by coordination of Computer Numerically Controlled (CNC) machines or 3D printers through a network and driven by computer programmed commands, rather than being controlled by hand. Such extensive use and reliance on IoT devices and software systems invites new cyber-physical threats. Due to the wide range of IoT-based manufacturing processes, this paper only concentrates on subtractive and additive pro-cesses, which serve as representatives of a larger group due to the fact that they are currently being used heavily in IoT environments[1].

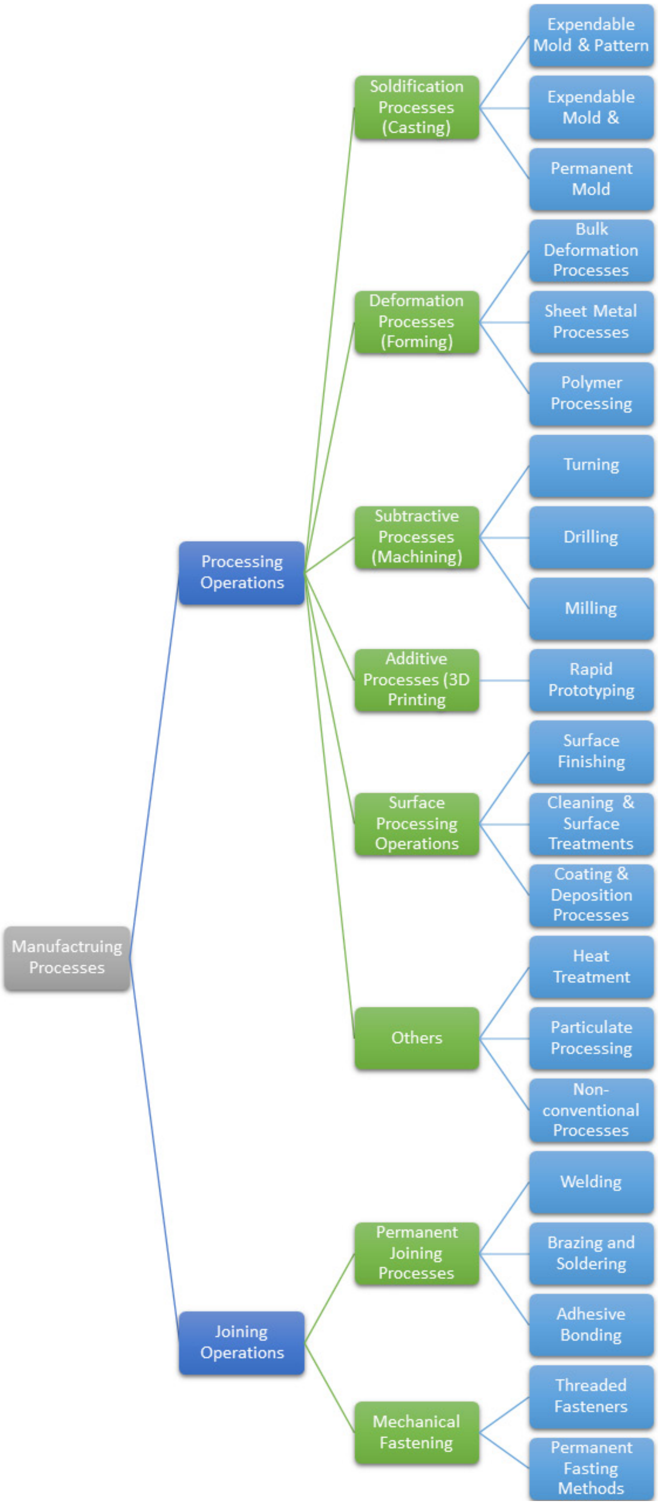

Manufactruing Processes
Processing Operations
Joining Operations
Soldification Processes (Casting)
Deformation Processes (Forming)
Subtractive Processes (Machining)
Additive Processes (3D Printing
Surface Processing Operations
Others
Permanent Joining Processes
Mechanical Fastening
Expendable Mold & Pattern
Expendable Mold &
Permanent Mold
Bulk Deformation Processes
Sheet Metal Processes
Polymer Processing
Turning
Drilling
Milling
Rapid Prototyping
Surface Finishing
Cleaning & Surface Treatments
Coating & Deposition Processes
Heat Treatment
Particulate Processing
Non-conventional Processes
Welding
Brazing and Soldering
Adhesive Bonding
Threaded Fasteners
Permanent Fasting Methods

Fig. 4. Manufacturing Processes Classification with Examples.

While subtractive and additive processes are significantly different, their integration into an IoT-based manufacturing system is relatively

[1] The attack taxonomy presented in Section II.C can also be applied to other manufacturing processes.

similar. Fig. 5 shows modern process chains for both an additive and a subtractive process, respectively. The process chain starts with a 3D CAD model, which is the digital representation of the shape and dimensions of an artifact.

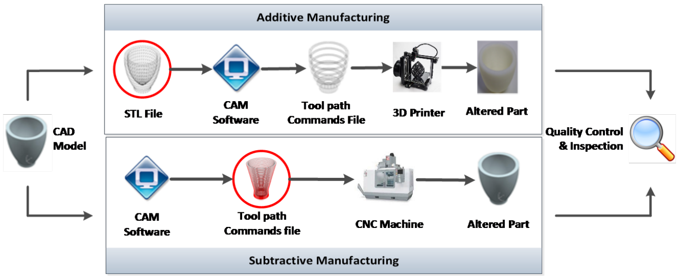


Fig. 5. Process Chains for Subtractive and Additive Manufacturing.

For subtractive manufacturing, the 3D CAD model goes directly to CAM software as modern CAD/CAM systems are integrated. After the CAM step is completed, a generic toolpath file is generated and sent to the IoT machine's controllers. In the process chains shown in Fig. 1, users have ready access to the toolpath, which provides a set of instructions for the tool regarding its direction, speed, and path.

In additive manufacturing, the CAD model is usually translated into an intermediary file format called an "STL" file, which represents the solid geometry with a list of triangu-lar facets that define a part's surface. Using machine-specific CAM software, this STL file is virtually sliced into layers that will be printed. Another algorithm generates commands (such as G-Code) that determine the additive manufacturing machine-specific toolpath to process each layer. This toolpath is generated locally on the machine or sent to a 3D printer's controllers across a network. These IoT systems allow designers to remotely print and monitor progress of different parts across the Internet.

In IoT-based manufacturing, each component of these process chains are linked through the IoT infrastructure, which poses potential risks of external cyber-physical attacks. In fact, two case studies [2], [3] conducted recently at Virginia Tech showed how to target a different component in each chain, as highlighted red in Fig. 5. In the case of the additive manufacturing process, a cyber-physical attack modified the STL file to create a part with an internal void [3]. In the case of the subtractive manufacturing process commands in the machine toolpaths were altered, thereby producing an incorrect part [2].

Examining the process chains of both additive and subtractive manufacturing demonstrates how vulnerable modern manufacturing is to cyber-physical attacks, *e.g.*:

- Both the STL and toolpath files are plain text without any encryption or encoding, which means these files can be intercepted and tampered/replaced. By modifying these files, attackers can bring parts out of specifications, add undesired part features, or alter part mechanical properties.
- An attack can propagate through an entire process chain. For example, altering a CAD file in transit across a network between IoT components will result in changes in the translated STL/toolpath file. If attacks cannot be prevented in previous processes, any quality inspection measures in later processes are meaningless.

C. A Taxonomy of Cyber-physical attacks against Manufacturing Processes

In this sub-section, we describe possible types of cyber-physical attacks against manufacturing system processes in IoT environments. An attack can be characterized by an *attack flow* where attackers first probe for a cyber vulnerability within the system, then exploit it with an appropriate attack vector to target a specific component within the manufacturing environment, producing a corresponding impact in form of a resulting attack. An example of an attack vector can be seen highlighted in red in Fig. 6, which also shows they key elements within an attack flow that could be described as follows:

(1) **Vulnerabilities** in the IoT-based manufacturing system can include a com-promised worker, OS/Software vulnerability, or weak authen-tication mechanism.

(2) **Attack Vectors** refer to paths where attackers can gain unauthorized access to the IoT system. Possible attack vectors include social engineering, malware like viruses or Trojans, insufficient authentication (attackers can get permission by brute force or bypass authentication), etc.

(3) **Attack Targets** are the actual assets (cyber or physical) being targeted by the attack. They can be manufactured products, the IoT machines used for manufacturing, or intellectual property, such as CAD design files or specifications.

(4) **Attack Impacts** result in different possible attack types, depending on the attack target. Those can be classified into three categories:

- **Confidentiality attacks** compromise the intellectual property of files, such as design model files. Design models may be highly confidential since they represent valuable business secrets for manufacturing companies. If these files are stolen by competitors and are used to reproduce similar products, substantial economic loss can be incurred for the company.
- **Availability attacks** affect the availability of manufacturing resources as they target manufacturing machines and tools. These attacks could deliberately slow down manufacturing processes by breaking down the controlling computers or damaging the manufacturing machines.
- **Integrity attacks** tamper with design models or configuration files of a manufacturing product line, thereby changing the geometric dimensions or mechanical properties of a part so it does not meet its designed requirements.

Based on the attack target, *integrity attacks* can be further categorized into *material attacks* and *structure attacks*, which are all shown in Fig. 6.

Material attacks are attacks that change the physical properties, such as material strength, surface roughness, color or magnetism of the manufacturing parts.

Structure attacks can change the following four types of geometric dimensions of manufacturing parts, as illustrated in [3]:

1. Scaling: a part is scaled up or down in one or more dimensions, resulting in various outcomes. For example, the part may no longer fit into other components or the part's mechanical properties may change by decreasing its strength.
2. Vertex movement: Some vertexes of a part have moved, which may not always change the part's external dimensions but alters the coordinates of certain vertexes internally. Vertex movement could result in fit issues or a change of mechanical properties.
3. Indents/protrusions: small indents or protrusions can be created on the surface of a part, resulting in fit issues or rough surface finish.
4. Internal void: a small volume created inside a part is not easily detectable by visual inspection since the void is completely enclosed. The void does not change the dimensions of a part. The void can impact a part's mechanical properties, e.g., if placed in a load bearing location, the void can make the part fail more easily. Additive manufacturing can create internal voids due to its layer-by-layer building process, but subtractive

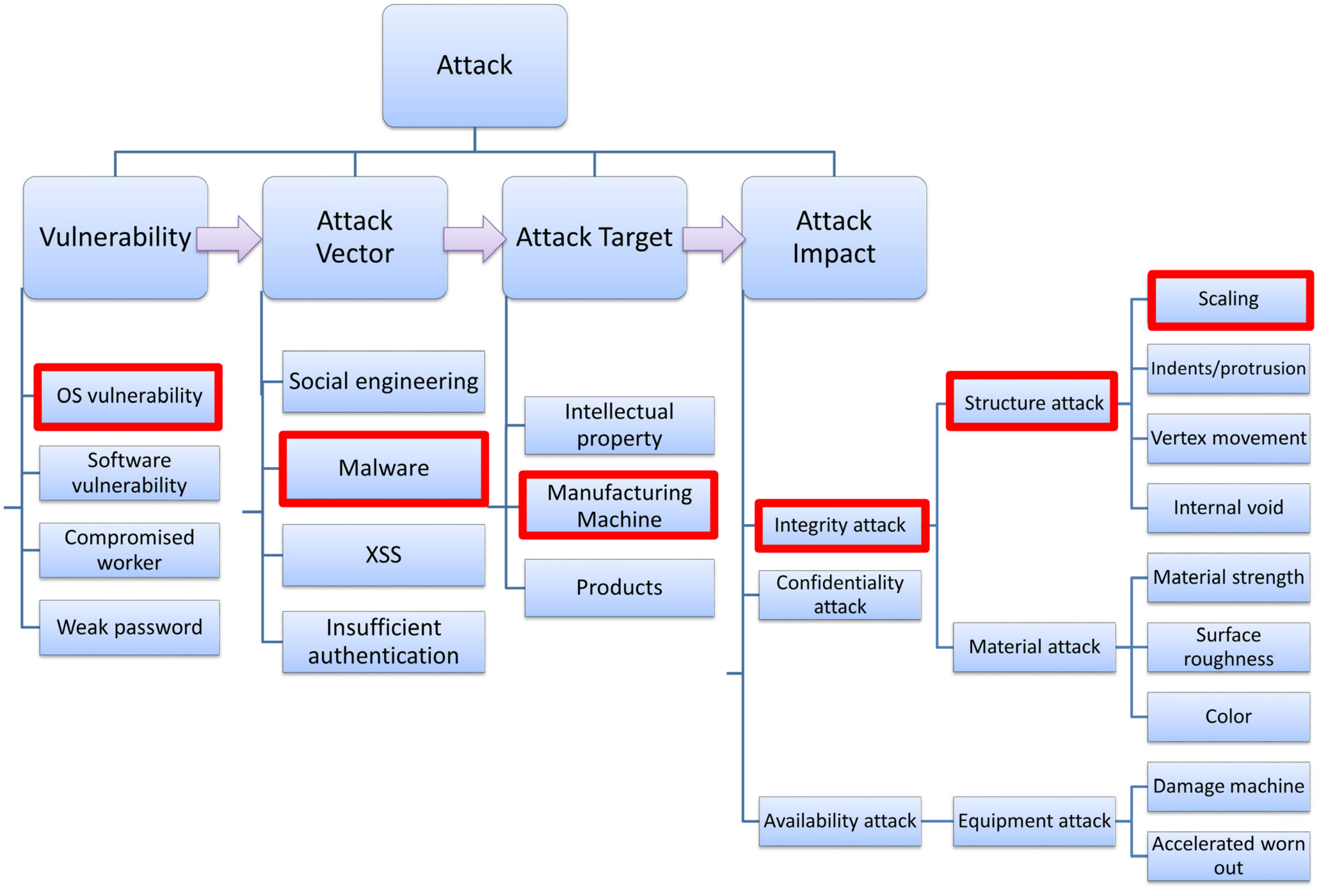


Fig. 6. Taxonomy of Cyber-Physical Attacks on Manufacturing Systems.

manufactur-ing cannot create internal voids.

Availability attack include *Equipment attacks* that aimed at IoT-based manufacturing equipment. For example, attackers can change machine configurations to force the equipment to operate outside its tolerance, causing damage to the machine, or accelerating wear and tear on the machine.

As shown with the red path in Fig. 6, an attack exploiting a vulnerability within the operating system, can apply a malware to target a manufacturing machine used in the facility. For instance, a structure attack could then affect the integrity of the machine through scaling the final product dimensions.

There is a relation between IoT-based manufacturing processes and cyber-physical attacks. Some attacks are only possible with the presence of certain manufacturing processes. For example, as previously mentioned, subtractive manufacturing processes, such as milling or turning generally, cannot create internal voids in manufactured parts. In contrast, 3D printing's flexibility makes it vulnerable to many kinds of at-tacks, including internal void attacks.

Table I presents a mapping between common manufacturing processes and their corresponding potential attack types. These relationships enable us to narrow down the possible attack types based on the manufacturing processes being used. In other words, after the desired attack type is determined, we can identify which specific manufacturing process would be affected; or we might even realize that the chosen attack type would not be possible and needs to be altered.

TABLE I. MANUFACTURING PROCESSES AND THEIR POTENTIAL ATTACKS.

Manufacturing Process	Vulnerability to Attack Types
Milling	Scaling, indents/protrusion, vertex movement, surface roughness
Turning	Scaling, surface roughness
Drilling	Scaling, indents/protrusion, vertex movement, surface roughness
3D printing	Scaling, indents/protrusion, vertex movement, internal void, material strength, color
Soldering	Material strength
Heat treatment	Material strength
Surface finishing	Color, surface roughness

D. A Taxonomy of Quality Inspection in IoT-based manufacturing Processes

We now present a second taxonomy of the quality inspection measures for manufacturing processes. Quality inspection "are measures aimed at checking, measuring, or testing of one or more product characteristics and to relate the results to the requirements to confirm compliance" [35]. It is an indispensable component in modern manufacturing to ensure products meet their quality requirements. Various quality inspection measures exist, each with its own pros and cons. For example, dimension measurement can detect scaling attacks, though it is ineffective against mechanical property attacks. It should just be noted that this quality inspection taxonomy has been developed assuming that the digital quality inspec-tion tools are not victims of cyber-physical attacks themselves; cyber-physical attacks compromising quality inspec-tion tools are beyond the scope of this paper.

Fig. 7 shows our taxonomy of quality inspection for manufacturing processes. These quality inspection measures can be applied to either the physical or the cyber domains of IoT-based manufacturing. The measures applied to the phys-ical domain usually measure the physical or mechanical properties of manufacturing parts to assess whether the de-sired requirements have been met. Based on the measured properties, quality inspection measures are usually non-destructive and can be classified into three groups: phys-ical characteristics, mechanical properties, and side-channel impacts. quality inspection measures for physical characteristics include visual inspection, dimension measure, weight measure, 3D laser scanning, X-rays, and CTs.

Mechanical properties refer to how parts behave under load. Mechanical properties include, but not limited to, strength (the resistance of a material to deformation from an external load), elasticity (the ability of a material to return to its original shape after the load is removed), and hardness (the ability of a material to resist indentation and scratching) [14]. These properties cannot be visually inspected, so tests must be run with specialized equipment to analyze these aspects of a part.

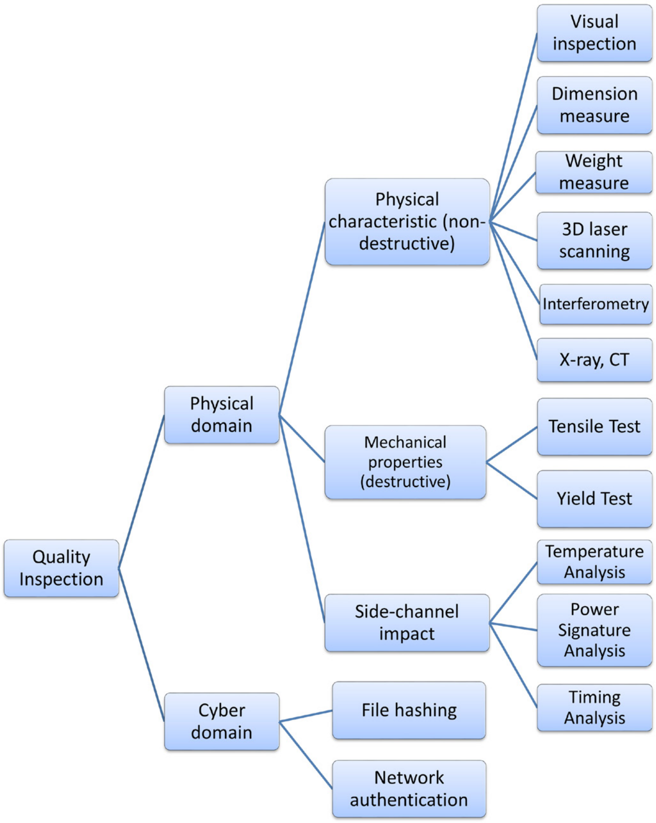


Fig. 7. A Taxonomy of Quality Inspection in Manufacturing Processes.

Side-channel impacts are mostly discussed in cryptography and refer to cases where attackers do not leverage information from plaintext or ciphertext, but from physical characteristics of cryptosystems. For instance, hardware has varying power consumption when doing different computations, such as adding and multiplying. By observing the power consumption of a cryptosystem, it is possible to deduce the key bits of RSA [19] or even to break the key [20]. Some other side-channel impacts include timing delays [20], electromagnetic leaks [21], temperature [22], or radiation [21]. Quality inspection in IoT-based manufacturing can measure side-channel impacts as well, to determine if a manufacturing process deviates from its designed specifications.

Quality inspection measures could be also combined with statistical analysis techniques since these tests may be expensive, destructive, or time consuming. Typical statistical analysis techniques are employed based on statistical models, in-cluding Statistical Process Control (SPC) [15], Six Sigma [16], acceptance sampling (where samples are chosen and analyzed in place of every part) [17], [18], etc.

In a way, the characteristics being measured are the parts facets, since it relates to its performance. Depending on such characteristics, linking quality inspection measures with the attack types described in Section II. C is important and can help determine which measures are effective against different attack types. A subset of the correspondences is shown in Table II. Again, cyber-physical attacks on quality inspection tools are not considered here.

TABLE II. CYBER-PHYSICAL ATTACK TYPES AND THEIR QUALITY INSPECTION MEASURES.

Attack Type	Effective Quality Inspection Measure
Scaling	Dimension Measure - Coordinate measure machine
Vertex movement	Dimension Measure - Coordinate measure machine
Indents/protrusion	Visual inspection
Internal void	X-ray, CT, side-channel information
Material strength	Tensile/yield strength test

E. Deducing Attack Threats from Software Vulnerabilities

A common misconception in the cyber-security community is that attacks can be avoided by simply employing the latest software versions and best practices. However, many IoT systems such as manufacturing equipment have long lifetimes, prohibitively high upgrade costs and need to remains operational continuously, and therefore cannot be migrated to the latest operating systems or manufacturing software versions. A key challenge, therefore, is to protect a complex IoT-based manufacturing process built on equipment with buggy or outdated software that cannot be easily upgraded to newer and more secure versions.

The cyber infrastructure refers to the computing equipment controlling physical manufacturing processes. Each computer equipment has several characteristics, such as operating system version (Windows XP, Windows 7, etc.), manufacturing software version (CAD, CAM software), and network connectivity status (Internet, LAN or None). The characteristic of the computers can be mapped to the exploitability vectors of vulnerabilities. A vulnerability with an access vector of “Internet” will only affects computers with an Internet connection.

To determine what attacks could be launched with known cyber vulnerabilities within the cyber infrastructure and what quality inspection measures should be taken to detect possible attacks, we have connected our attack taxonomy with the National Vulnerability Database (NVD) [23]. The NVD is a U.S. government repository of vulnerability management data, which uses the Common Vulnerability Scoring System (CVSS) [24] to evaluate the severity of vulnerabilities. The CVSS defines a set of metrics to describe the characteristics of vulnerabilities. The metrics includes six vectors that are described below. The first three of these vectors in CVSS are organized in terms of exploitability:

- **Access Vector** (AV) measures an attacker’s ability to successfully exploit a vulnerability based on how remote an attacker can be from a networking perspective [25]. There are three possible values for Access Vector: Local, Adjacent Network, and Network. An Access Vector of value “Network” (AV: N) means the vulnerability must be exploitable without requiring physical (*i.e.*, local) or adjacent network access. Often, AV: N vulnerabilities can be exploited from IP addresses on the Internet. An Access Vector of value "Adjacent Network" means the vulnerability must be exploitable through a broadcast or collision domain. An Access Vector of value "Local" means the vulnerability must only be exploitable via physical access, such as proximity to a device or local shell access.

- **Access complexity** measures the complexity of the attack required to exploit the vulnerability after the attacker gained access to the target system already [25].
- **Authentication** measures the number of times an attacker needs to authenticate to the target system to exploit a vulnerability [25].

The Access Complexity and Authentication vectors describe the degree of difficulty, but not possibility of an attack, which are not relevant to our taxonomy, so we omit their discussions here.

Three other vectors in CVSS are organized in term of impact:

- The **Confidentiality Metric** measures the attacker's ability to obtain unauthorized access to information from an application or system [25]. If no information or data is exposed due to exploitation, the Confidentiality metric receives a value of "None". If only partial information is disclosed due to exploitation (the attacker cannot control what is obtained), the Confidentiality metric receives a value of "Partial". If an attacker has complete read access to all information and data on a system, the Confidentiality metric receives a value of "Complete". The compromise of confidentiality metric means the vulnerability can help attackers gain "read" access to the system. The "read" access will make it possible to launch confidentiality attacks that are discussed in Section II.C.
- The **Integrity Metric** measures an attacker's ability to manipulate or remove data from a product or system [25]. There are three possible values for this metric: None (I: N), Partial (I: P), and Complete (I:C). "None" is used when vulnerability exploitation cannot manipulate data. For example, an information leak only exposes information but unauthorized modification is not possible. A "Partial" impact to Integrity implies limited or uncontrolled modifications to files are possible by exploiting a vulnerability. An Integrity metric of "Complete" means an attacker is able to modify any system files or data in the system. The compromise of integrity metric means the vulnerability can help attackers gain "write" access to the system. The "write" access will make it possible to launch integrity attacks. Integrity attacks usually need to change the critical part of design files or machine configurations, a "partial" impact is not sufficient because attackers cannot make predictable changes. The partial value is therefore treated the same as no value.
- The **Availability Metric** measures an attacker's ability to disrupt or prevent access to services or data [25]. Vulnerabilities can impact availability by affecting hardware, software, and network resources. For example, vulnerabilities can make it possible for attackers to flood network bandwidth, exhaust CPU or system memory. There are three possible values for this metric: None (A: N), Partial (A: P), and Complete (A: C). The compromise of availability metric means it is possible to launch availability attacks.

We now examine some vulnerabilities from the NVD to see how they can be connected to our proposed taxonomy. As shown in Table III, CVE-2014-7268 is a vulnerability whose description is "*Buffer overflow in AClient in Symantec Deployment Solution 6.9 and earlier on Windows XP and Server 2003 allows local users to gain privileges via unspecified vectors*." As shown in Table III, the prerequisites of vulnerability CVE-2014-7268 are installations of Symantec Deployment Solution on Windows XP or Server 2003 operation systems and local access to the computers involved in the IoT-based manufacturing process. If these prerequisites are met, this vulnerability can be exploited to launch attacks that result in "complete" confidentiality, integrity, and availability impacts, which means all the attacks in our taxonomy shown in Fig. 6 could be launched by exploiting this vulnerability.

TABLE III. EXAMPLE VULNERABILITY AND METRICS.

Vulnerability	Metric	Value
CVE-2014-7286	Vulnerable software	Symantec Deployment Solution 6.9 or earlier on Windows XP or Windows server 2003
	Access vector	Local
	Confidentiality	Complete
	Integrity	Complete
	Availability	Complete
CVE-2015-2453	Vulnerable software	Windows vista, 7, 8, 8.1, server 2008, 2012
	Access vector	Local
	Confidentiality	Complete
	Integrity	None
	Availability	None

Table III also shows the metrics for vulnerability CVE-2015-2453, which is documented as "*The Client/Server Run-time Subsystem (CSRSS) in Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 allows local users to obtain sensitive information via a crafted application that continues to execute during a subsequent user's login session, also known as "Windows CSRSS Elevation of Privilege Vulnerability"*." This vulnerability just impacts confidentiality, so only confidentiality attacks can be launched and manufacturers need not prepare for integrity attacks or availability attacks. Moreover, manufacturers need not do anything if the manufacturing design files are publically available, *i.e.*, intentionally not confidential.

III. Case Study

An increasing number of manufacturing companies have embraced the Internet of Things to revolutionize the way they manufacture. Information technology infrastructure has been used extensively in design, manufacturing processes and quality inspection for accessing the information of physical objects and for manipulating them. The tight integration of hardware and software enables a more efficient production management. While modern manufacturing companies are enjoying the benefits the IoT brings, most of them are unaware of the potential cyber-security risks they may face.

To demonstrate how our taxonomies can be applied to modern manufacturing systems to assess cyber-security risks, we visited an industry partner to collect related information and map them to our approach. This company provides additive manufacturing services that allow customers to submit their own parts designs to facilitate production.

The general process flow of this company is shown in Fig. 8. A customer submits parts through a web portal or directly through email to a product engineer, who then coordinates with the customer to determine the printability and best material/process. The part files (in CAD or STL format) will be saved to the network drive. The process engineer checks the file for common problems, such as thin walls or extra shells, and adjusts the files if necessary. Machines will also be checked before printing. After that, the parts will be printed (along with witness bars) and will go through quality inspection measures. If the parts pass inspection, they will be shipped to customers; otherwise, they will be scrapped or reworked.

The Information Technology (IT) infrastructure in this manufacturing company consists of three categories of computers: engineers' computers, 3D printer computers, and inspection station computers. Files are stored on a networked server connected to all computers. There are no restrictions on USB drives and all computers have USB access. No personal computers are allowed, but work laptops can be taken home and can remotely access the server. Many computers run outdated operating systems, including Windows XP and Windows 7. Most computers are connected with the Internet to access the design files from network drive. For computers without the need to access design files, many cannot be unplugged due to the restriction of Digital Rights Management (DRM) systems or software activation.

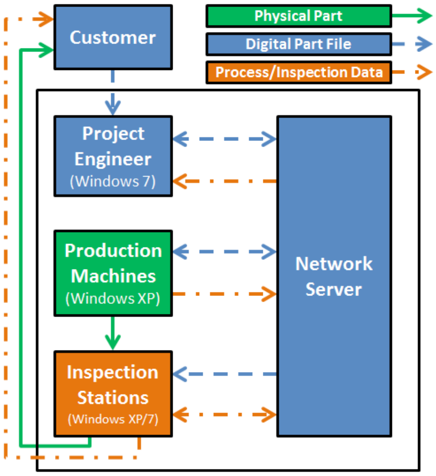

Customer
Physical Part
Digital Part File
Process/Inspection Data
Project Engineer (Windows 7)
Production Machines (Windows XP)
Inspection Stations (Windows XP/7)
Network Server

Fig. 8. General Process Layout of the Manufacturing Company.

This company applies many quality inspection measures, including digital file checks, machine process checks, material quality checks, and part quality checks. Digital file checks verify the STL file and determine if there are any inverted normals, holes, or non-closed shells. Machine process checks include assessing laser power, IR sensor, or O2 sensor to ensure the machine is operating normally. Material quality checks includes checking the powder mix ratio and the melt flow index to see the powder batch being used meet the requirements. Part quality checks include dimension measure, visual inspection, and tensile test. Dimension measures are performed with Faro Arm (a portable coordinate measuring machine) and manually by calipers.

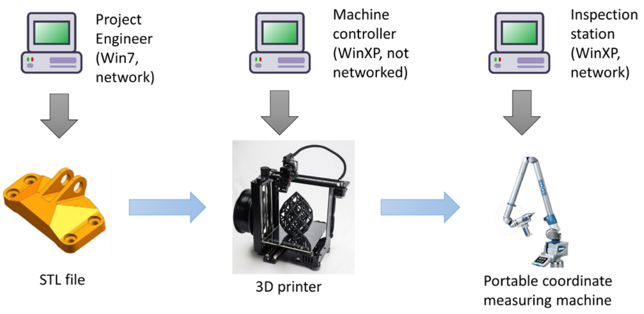

Project Engineer (Win7, network)
Machine controller (WinXP, not networked)
Inspection station (WinXP, network)
STL file
3D printer
Portable coordinate measuring machine

Fig. 9. An Example Product Line.

We applied our taxonomies to conduct a systematic risk assessment for this manufacturing company. Fig. 9 shows an example product line that consists of a single process: *3D printing*. Vulnerability "CVE-2015-2453" presented in Section II.D is an operating system vulnerability that will impact all computers running Windows 7 with "complete" confidential impact. Since the project engineer's computer is running Windows 7 and the STL file is stored in this computer, the vulnerability will allow attackers to launch confidentiality attacks to steal the design files.

Vulnerability CVE-2014-7268 will impact all the computers running Symantec Deployment Solution 6.9 or earlier on Windows XP with "complete" confidentiality/integrity/availability impact, which means attackers could launch integrity attacks by gaining write access to computers controlling 3D printer and inspection station.

IV. Related Work

Prior work has explored various types of security issues in cyber-physical systems. For example, Cardenas et al. [26] discuss key challenges for securing cyber-physical systems and Sridhar et al. [27] model the security risks for the Electric Power Grid. However, they do not consider the domain knowledge of manufacturing in their security models.

Taxonomies have been proposed for cyber-attacks in Information Technology (IT) systems [28], [29]. While the taxonomies are useful for manufacturing systems to defend traditional cyber-attacks, these taxonomies do not capture the physical effects of the attacks on IoT-based manufacturing systems. In IoT-based manufacturing systems, the attacks on the controlling systems can directly impact the physical world.

Taxonomies have also been proposed for cyber-attacks in the IoT systems. For example, Zhu et al. [6] analyze the cyber-attacks on Supervisory Control and Data Acquisition systems. No equivalent taxonomy has been proposed, however, to systematical classify possible cyber-physical attacks in manufacturing systems. However, no equivalent taxonomy has been proposed in manufacturing.

Integrated circuit manufacturing faces similar security challenges as cyber-physical manufacturing systems [30]. Taxonomies have been developed for hardware Trojans [7], [8], [30], which are maliciously injected logic in integrated circuits. Tehranipoor et al. [7] survey the design and taxonomy of hardware Trojan. Detection methodologies for hardware Trojans are also discussed in their survey. Jin et al. [8] present different implementations of hardware Trojans and show that traditional functional testing can be useless in detecting hardware Trojans.

Quality inspection in integrated circuits aims to detect if a manufactured circuit matches its original design [8]. Since circuits cannot be easily deconstructed for testing, side-channel detection is widely used as a quality inspection measure for defending against hardware Trojans. Researchers have developed various side-channel methods including timing delays [31], power analysis [32] for detecting hardware Trojans. Cyber-physical attacks in manufacturing systems differ from hardware Trojan in that the manufactured parts are not electronic in nature and there is no computational logic to verify the functions; yet some similarities could still exist as discussed in [33].

Hence, taxonomies to systematical classify possible cyber-physical attacks in manufacturing systems and provide a framework to reason about the relationship between attack types, processes, equipment and quality inspection measures were needed.

V. Concluding Remarks

The Internet of Things (IoT) has transformed many aspects of modern manufacturing. IoT-based manufacturing systems, however, are much more vulnerable to cyber-physical attacks than traditional manufacturing systems. Given the importance of IoT-based manufacturing systems throughout the supply chains in modern economies, identifying and remediating these vulnerabilities is of paramount importance [34].

To understand potential dangers and protect manufacturing system safety, this paper presents two taxonomies: one for classifying cyber-physical attacks against IoT-based manufacturing processes and another for quality inspection measures for counteracting these attacks. These taxonomies provide guidance for evaluating IoT-based manufacturing system security by delineating the research space and helps to codify and relate research approaches to one another. These taxonomies also build connections between IoT-based manufacturing processes, attacks, and quality inspection measures.

Based on creating our taxonomies and applying them in the context of the case study in Section III, we have identified the following lessons learned:

- Manufacturing companies can benefit from these taxonomies to reason more effectively about what possible attacks could happen to their IoT-based manufacturing process chains, as well as ascertain which quality inspection measures are needed to detect defects resulting from cyber-attacks on IoT-based manufacturing infrastructure.
- Ensuring the security of IoT-based manufacturing systems is a cross-disciplinary problem that can be solved most effectively by collaborative efforts of researchers from both cyber-security and mechanical engineering domains. Moreover, knowledge of cyber-security should be explained in manufacturing terms to enable meaningful reasoning.
- There is a tradeoff between quality inspection measure coverage and the costs. Enforcing more quality inspection measures can examine more aspects of the products, but with a higher cost. Our taxonomies can help eliminate quality inspection measures that are not necessary and prioritize quality inspection measures that ensure quality attributes that requirements manufacturers value the most.

Now that we have created these taxonomies, our next step is to develop an analysis tool to emulate current IoT-based manufacturing systems. Given IoT-based manufacturing process structures, system configurations, and budgets, this analysis tool will provide quality inspection recommendations on where and how to test. We also plan to explore what side-channel information can be utilized to detect attacks and develop algorithms to detect attacks by processing side-channel data in IoT-based manufacturing processes.

ACKNOWLEDGE

This work was supported in part by the National Science Foundation and Department of Homeland Security through Grant #CNS1446303.

References

[1] H. Turner, J. White, J. A. Camelio, C. Williams, B. Amos, and R. Parker, “Bad parts: Are our manufacturing systems at risk of silent cyberattacks?” *IEEE Security & Privacy,* no. 3, pp. 40–47, 2015

[2] L. J. Wells, J. A. Camelio, C. B. Williams, and J. White, “Cyber-physical security challenges in manufacturing systems,” Manufacturing Letters, vol. 2, no. 2, pp. 74–77, 2014.

[3] L. Sturm, C. Williams, J. Camelio, J. White, and R. Parker, “Cyberphysical vulnerabilities in additive manufacturing systems,” 25th Annual Solid Freeform Fabrication Symposium, vol. 7, p. 8.

[4] S. Hurd, C. Camp, J. White, Quality Assurance in Additive Manufacturing Through Mobile Computing, The 7th EAI International Conference on Mobile Computing, Applications and Services, Nov 12-13, 2015, Berlin, Germany.

[5] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” Security & Privacy, IEEE, vol. 9, no. 3, pp. 49–51, 2011.

[6] B. Zhu, A. Joseph, and S. Sastry, “A taxonomy of cyber attacks on scada systems,” in Internet of things (iThings/CPSCom), 2011 international conference on and 4th international conference on cyber, physical and social computing. IEEE, 2011, pp. 380–388.

[7] M. Tehranipoor and F. Koushanfar, “A survey of hardware trojan taxonomy and detection,” 2010.

[8] Y. Jin, N. Kupp, and Y. Makris, “Experiences in hardware trojan design and implementation” IEEE International Workshop on Hardware-Oriented Security and Trust. IEEE, 2009, pp. 50–57.

[9] M. P. Groover, Fundamentals of modern manufacturing: materials processes, and systems. John Wiley & Sons, 2007.

[10] E. P. De Garmo, J. T. Black, and R. A. Kohser, DeGarmo’s materials and processes in manufacturing. John Wiley & Sons, 2011.

[11] S. Kalpakjian and S. R. Schmid, Manufacturing, Engineering and Technology 7th Edition. Pearson Education, Inc., 2014.

[12] Oberg, Erik, et al. Machinery's handbook. Vol. 200. New York: Industrial Press, 2004.

[13] Schmidt, Douglas C. "Model-Driven Engineering." IEEE Computer, 39.2 (2006): 25.

[14] R. W. Messler and R. W. Messler Jr, The essence of materials for engineers. Jones & Bartlett Publishers, 2010.

[15] J. S. Oakland, Statistical process control. Routledge, 2007.

[16] M. J. Harry and R. R. Schroeder, Six Sigma: The breakthrough management strategy revolutionizing the world’s top corporations. Broadway Business, 2005.

[17] E. G. Schilling, Acceptance sampling in quality control. CRC Press, 1982.

[18] Montgomery, Douglas C. Introduction to statistical quality control. John Wiley & Sons, 2007.

[19] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” in Advances in cryptology. Springer, 1985, pp. 10–18.

[20] P. C. Kocher, “Timing attacks on implementations of Diffie-hellman, RSA, DSS, and other systems,” in Advances in CryptologyCRYPTO96. Springer, 1996, pp. 104–113.

[21] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, “Side channel cryptanalysis of product ciphers,” in Computer Security ESORICS 98. Springer, 1998, pp. 97–110.

[22] M. Hutter and J.-M. Schmidt, “The temperature side channel and heating fault attacks,” in Smart Card Research and Advanced Applications. Springer, 2014, pp. 219–235.

[23] “Nist vulnerability database,” https://nvd.nist.gov/

[24] P. Mell, K. Scarfone, and S. Romanosky, “Common vulnerability scoring system,” Security & Privacy, IEEE, vol. 4, no. 6, pp. 85–89, 2006.

[25] J. Franklin, C. Wergin, and H. Booth, “CVSS implementation guidance,” National Institute of Standards and Technology, NISTIR-7946, 2014.

[26] Cardenas, Alvaro, et al. "Challenges for securing cyber physical systems."Workshop on future directions in cyber-physical systems security. 2009.

[27] Sridhar, Siddharth, Adam Hahn, and Manimaran Govindarasu. "Cyber–physical system security for the electric power grid." Proceedings of the IEEE 100.1 (2012): 210-224.

[28] Hansman, Simon, and Ray Hunt. "A taxonomy of network and computer attacks." Computers & Security 24.1 (2005): 31-43.

[29] Killourhy, Kevin S., Roy A. Maxion, and Kymie Tan. "A defense-centric taxonomy based on attack manifestations." 2004 International Conference on Dependable Systems and Networks. IEEE, 2004.

[30] Wang, Xiaoxiao, Mohammad Tehranipoor, and Jim Plusquellic. "Detecting malicious inclusions in secure hardware: Challenges and solutions." Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on. IEEE, 2008.

[31] Y. Jin and Y. Makris, “Hardware Trojan detection using path delay fingerprint,” in IEEE International Workshop on Hardware-Oriented Security and Trust (HOST 2008), 2008, pp. 51– 57.

[32] R. Rad, J. Plusquellic, and M. Tehranipoor, “A sensitivity analysis of power signal methods for detecting hardware Trojans under real process and environmental conditions,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 12, pp. 1735–1744, 2010.

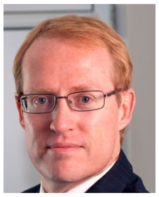
[33] Vincent, Hannah, et al. "Trojan Detection and Side-channel Analyses for Cyber-security in Cyber-physical Manufacturing Systems." Procedia Manufacturing 1 (2015): 77-85.

[34] CERT-UK, “Cyber-security Risks in the Supply Chain,” available from https://www.cert.gov.uk/wp-content/uploads/2015/02/Cyber-security-risks-in-the-supplychain.pdf.

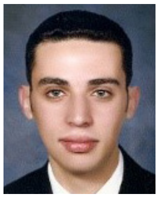
[35] CEOPEDIA, https://ceopedia.org/index.php/Quality_inspection, 2016.

Yao Pan received the B.S. degree in Computer Science in June 2012, from Zhejiang University in China. He is currently working toward the Ph.D. degree under the supervision of Dr. Jules White, at the Department of Electrical Engineering and Computer Science, Vanderbilt University. His research interests include cyber security, distributed systems, cloud computing and mobile computing.

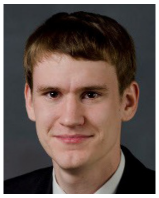
Jules White is an Assistant Professor of Computer Science in the Department of Electrical Engineering and Computer Science at Vanderbilt University. He is a National Science Foundation CAREER Award recipient. Dr. White's research focuses on securing, optimizing, and leveraging data from mobile cyber-physical systems. His mobile cyber-physical systems research spans four key focus areas: (1) mobile security and data collection, (2) high-precision mobile augmented reality, (3) mobile device and supporting cloud infrastructure power and configuration optimization, and (4) applications of mobile cyber-physical systems in multi-disciplinary domains, including energy-optimized cloud computing, smart grid systems, healthcare/manufacturing security, next-generation construction technologies, and citizen science. His research has been licensed and transitioned to industry, where it won an Innovation Award at CES 2013, attended by over 150,000 people, was a finalist for the Technical Achievement at Award at SXSW Interactive, and was a top 3 for mobile in the Accelerator Awards at SXSW 2013.



Douglas C. Schmidt is a Professor of Computer Science at Vanderbilt University. His research covers a range of software-related topics, including patterns, optimization techniques, and empirical analyses of object-oriented middleware frameworks for distributed real-time embedded systems and mobile cloud computing applications. Dr. Schmidt received B.S. and M.A. degrees in Sociology from the College of William and Mary in Williamsburg, Virginia, and an M.S. and a Ph.D. in Computer Science from the University of California, Irvine (UCI) in 1984, 1986, 1990, and 1994, respectively.



Ahmad Elhabashy rreceived B.S. and M.Sc. degrees in Production Engineering in 2009 and 2012 respectively, from Alexandria University, Egypt. He is currently working towards a Ph.D. degree under the supervision of Dr. Jaime Camelio at the Grado Department of Industrial and Sys-tems Engineering, Virginia Tech. His research interests include Quality control, production planning and control, modeling of industrial systems and optimization particularly in manufacturing context.



Logan Sturm graduated from Virginia Tech with a Bachelor of Science in Mechanical Engineer in 2013. As an undergraduate he performed research in the MicrON Lab on BacteriaBots, and served as captain and mechanical team lead for the Autonomous Underwater Vehicle Team. While in school Logan completed two years as a manufacturing engineering intern at Federal Mogul. For his senior capstone design project, Logan programed the software and user interface for a Mask Projection Micro-Stereolithography 3D Printer that he and his team designed and built. The team's final project received the award for "Best Design Project" from among 40 mechanical engineering capstone projects. Logan joined the DREAMS Lab in 2013 as an undergraduate researcher and is now pursuing a Ph.D. degree in Mechanical Engineering. Logan's current research is in Cyber-Physical Security for AM systems. This involves identifying current vulnerabilities in AM machines and in the process chain, and determining ways to detect and mitigate attacks.

Jaime Camelio is the Rolls-Royce Commonwealth professor for advanced manu-facturing at the Grado Department of Industrial and Systems Engineering at Virginia Tech. He leads the Virginia Tech Cyber-Physical Systems Security Manufacturing Group, which along with its industry partners and alliance with government agencies, is looking to improve the resiliency of the critical infrastructure of the United States, specifically the manufacturing related segments. Dr. Camelio holds a Ph.D. in Mechanical Engineering and a M.S. in Industrial Engineering from the University of Michigan and a B.S. and M.S. degrees in Mechanical Engineer-ing from the Universidad Catolica de Chile. His research interests are in assembly systems, intelligent manufacturing, process moni-toring and control, and cyber-physical security in manufacturing. He has authored or co-authored more than 70 technical papers and holds one patent.

Christopher Williams is an Associate Professor and the Electro-Mechanical Corporation Senior Faculty Fellow in the Department of Mechanical Engineering at Virginia Tech. He is the Director of the Design, Research, and Education for Additive Manufacturing Systems (DREAMS) Laboratory (DREAMS Lab), and the Associate Director of Virginia Tech's Macromolecules & Interfaces Institute. He holds affiliate faculty appointments in the Department of Engineering Education and the Department of Material Science & Engineering. His Additive Manufacturing (AM) expertise is focused in innovations in (i) AM processes and materials; (ii) design methodologies and tools to guide AM use; and (iii) AM workforce development initiatives. Dr. Williams has authored over 100 peer-reviewed articles and has presented 30+ invited talks. Dr. Williams is also a recipient of a National Science Foundation CAREER Award (2013). His research contributions have been recognized by eight Best Paper awards at international design, manufacturing, and engineering education conferences. Dr. Williams holds a Ph.D. and M.S. in Mechanical Engineering from the Georgia Institute of Technology and a B.S. with High Honors in Mechanical Engineering from the University of Florida.

Analysis of Security Mechanisms Based on Clusters IoT Environments

Paulo Gaona-García[1], Carlos Montenegro-Marin[1], Juan David Prieto[2], Yuri Vanessa Nieto[3]

[1]*Universidad Distrital Francisco José de Caldas*
[2]*Fundación San Mateo*
[3]*Corporación Unificada Nacional de Educación Superior, Bogotá – Colombia,*

***Abstract* — Internet of things is based on sensors, communication networks and intelligence that manages the entire process and the generated data. Sensors are the senses of systems, because of this, they can be used in large quantities. Sensors must have low power consumption and cost, small size and great flexibility for its use in all circumstances. Therefore, the security of these network devices, data sensors and other devices, is a major concern as it grows rapidly in terms of nodes interconnected via sensor data. This paper presents an analysis from a systematic review point of view of articles on Internet of Things (IoT), security aspects specifically at privacy level and control access in this type of environment. Finally, it presents an analysis of security issues that must be addressed, from different clusters and identified areas within the fields of application of this technology.**

***Keywords* — Internet of Things, Network Security, Information Security, Privacy of Data, Secure Connections.**

I. Introduction

Internet of things (IoT) is considered as an integrated part of Internet, also defined as a global network infrastructure and dynamic composed of a large number of objects, able to communicate and interact with each other, with end users [1][2][3]. These objects must have unique identities which allow interactivity.

Due to the accelerated enhanced of devices connected to Internet, and the need to create networks that interact with them, privacy and data protection is substantial [4]. Therefore, the information security is an actual well known aspect, due to devices connected to internet is growing rapidly, which represents an exposure increase on data at the network.

This paper proposes a security infrastructure to neutralize vulnerabilities at IoT, using mechanisms such as (PKI) that allow identity authentication based on a combined public key, giving solution to the excessive amount of authentications. In this issue it is found a solution given by [5] performing an analysis of fingerprint recognition, they proposed a 3-layer model (sensor, transport, application), enabling the analysis of each of the components involved in the process. Another security problem is related to the communication media, this problem is addressed in [4], in this project authors using RFID systems and incorporate a microchip combined memory, create a system which allows to receive a signal and return it with some additional data (unique serial number).

This study is intended to present an overview of challenges presents in IOT security levels. Thus, it is presented the state of the art related to safety in environments Internet of things, specifically about security mechanisms involved in it and on the other hand, present an analysis of factors involved in performance application and security, and identify security methods that allow be implemented in IOT environments. In order to achieve this purpose, we will introduce classification as a proposal to identify which aspects should be considered for raising safety issues under the principles of authentication, access control and authenticity. For this model, it is important to characterize the type of RFID devices, work settings, connection types and security mechanisms that could be applied for the purpose in order to facilitate the acquisition of devices to be used in different work environments such as industrial level, SmartGrid or home.

This paper is organized as follows. In Section 2, we presents the theoretical framework, problems identified in the area of security and related work. Section 3 describes the methodology for the literature review and analysis of our study is presented. Section 4 a proposed security model according to areas of interest to today worked in IoT is presented. The final section conclusions and future work is presented.

II. Background

A. Review Stage

Recently, Internet of Things (IOT), has become a trend at homes given the evolution and mass communications through the network, which facilitates the exchange of goods and services globally [6]. In accordance with [7] monitoring households through security cameras, motion detectors and other various sensors which are connected to the Internet allow to handle them easily, and the flow of valuable information for the user. These factors creates tranquility, for example, being able to monitor home from anywhere in the world having a smartphone connected to Internet. Nevertheless, in accordance with a study handle by [8] these constant monitoring levels are exposed to confident levels to analyze the risks, for example, the network points and transmission thereof. The authors finally conclude the need to review the processes of encryption and authentication of this. That means, now the user is not the only one who can see and monitor his home, but this would be a relatively easy task for an intruder, exposing and becoming the privacy and security of a house vulnerable.

Some of the most popular devices are leading the expansion of IoT are called wearables; They are small devices that can be wear by a person and can capture information from certain activities carried out. They can also provide other information to the user such as time, weather or even notifications received on the same or on a mobile phone linked. In addition to synchronize activity with other devices or social networks, they are able to receive mail, messages, and even calls, so in most cases the information is stored in the cloud.

IoT links computer systems to the real world through physical objects, which allow having real-time information [7]. This means that a lot of information should travel safely from objects (sensors, actuators, RFID tags, etc.), to the data center and from there to devices such as PC or smartphone, from they can make decisions based on the information it reaches. It is the development of IOT which brings new challenges in security aspects.

DOI: 10.9781/ijimai.2017.438

B. Related work

The rapid development of information technology and Internet security information about IoT, I a new problems and potential security over information has been rise. Therefore, it becomes a focus aspect to build a safety and reliability system in the IoT context. Form this problem, it has been worked in a general architecture of trust [9] this architecture mainly includes a trust module (users being the central part of the system) perception of trust module (full authentication) terminal confidence module (operate according to rules of control), trusted network module (designed to analyze, evaluate and manage security situations) and a trusted agent module (avoid the potential risks caused by access terminals do not reliable). According to the results of these modules development, was a development model to address security issues, but does not provide a specific solution to the security problem.

As the communications infrastructure of the Internet evolves to include detectable objects, appropriate mechanisms will be needed to ensure communications with these devices in the work done by [10] in the context of future applications of IoT, in areas as diverse as health (eg, remote patient monitoring or control of the elderly) and smart cities (eg distributed pollution monitoring, intelligent lighting systems), among many others. This trend is also reflected in the efforts carried out by normative agencies such as the Institute of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF), to design communication technologies and safety the IOT.

C. Identify security problems

The atmosphere of Internet of things can be summed up as a virtual representation in the network of physical objects. Data interception is real and possible, this can be proof thanks to studies that have managed to activate windshield wipers and brakes of cars only through text messages [11], handling electronic devices of the vehicle [12], tracking the vehicle navigation system [13], annulment of the navigation system of a luxury yacht running aground in the middle of the Mediterranean [14] sea, among others.

Security is a factor that should be taken into account from the start design of any product. Such problems could be trivial if other violations that have occurred at industrial sector where signals are forged through networks or wireless sensors are analyzed; but even more worrying when heating systems, lighting and security of households are tapped to be changed and transgressed [6] [7] [15] [16].

According to a study from Hewlett Packard [17] about 70% of Internet things devices are vulnerable to attack. Security cameras, thermostats, alarms, door controllers were studied, among others; each of these had a service oriented to the cloud and had a mobile application. About 25 vulnerabilities for each device and the following were highlighted:

i) Privacy issues (where you can delve respect to the rights inherent human beings to this principle).

ii) Insufficient authorization, iii) lack of encryption

iii) Insecure web interface

iv) Inadequate protection software.

The report of the most common threats in IoT [17] is presented in Fig. 1.

Hewlett Packard's report also highlighted that information such as credit cards, social security numbers and other sensitive data travel over the network without proper security. While this study has certain commercial purposes, it is important to identify issues facing the industry determines as relevant in this market.

On the other hand with the development of the IoT, RFID and ubiquitous network technology sensors have become two major parts of it. RFID as a type of automatic identification technology without contact identifies objects through RF signal and collect data. It is possible to work in different environments and identifying objects, according to [18] RFID is now often seen as a prerequisite for the IoT. In general, the IoT can be divided into three layers:

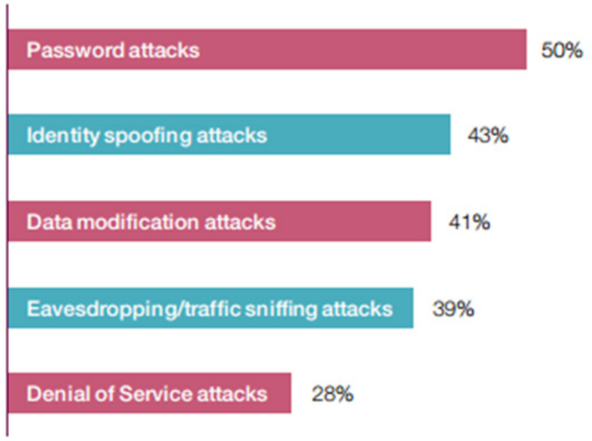


Fig. 1. Top the most common threats of IoT products (Capgemini consultant

- *The lower level,* is the perception layer used mainly to capture, gather, distinguish and identify object information. The layer includes RFID tags and literacy devices, cameras, GPS, sensors, laser scanner, and so on.
- *The second level is the network layer,* which is used to transmit and process information obtained by the layer of perception and provides such information to the application layer, with the support of reliable communication.
- *The upper level* is the application layer, used to process data intelligently, and aggregation of data from various sources with different types. The layer implements control and information management, making use of cloud computing, data mining etc.

This model provides a theoretical framework for building a reliable security information, enabling IoT to be a creditable, controllable and independent network.

As the communications infrastructure of the Internet evolves to include detectable objects, appropriate mechanisms will be needed to ensure communications with these devices in the work done by [10] in the context of future applications of IoT, in areas as diverse as health (eg, remote patient monitoring or control of the elderly) and smart cities (eg distributed pollution monitoring, intelligent lighting systems), among many others. This trend is also reflected in the efforts carried out by normative agencies such as the Institute of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF), to design communication technologies and safety the IOT. Such technologies currently form a stack of protocols required for IoT with various communication technologies; work done by [19] is discussed detailed.

Some applications such as a monitoring system houses by [20], works using open microcontrollers such as Arduino code. Arduino Atmel AVR uses a processor that can be programmed in C language computer through the USB port also allows you to interact with other devices. The Ethernet module acts as a bridge to connect the Home Gateway to the local proxy. This application consists of three main modules: Web micro server, hardware interface modules and software package (smartphone app). This work proposes the implementation of a new architecture for the surveillance system using Android-based smartphone ensuring low costs and home control flexibly. The proposed architecture uses Web services based on Representational State Transfer (REST), as a layer of interoperable communication between the remote user and the application home devices.

From these references, then the analysis work related work, it was classified aspects from clusters defined on IoT area.

III. Methodology

In order to make an analysis of literature review, we carry out three phases to identify related works. The first preliminary phase we used keywords in fields related to security issues in IoT environments in databases such as IEEE, ACM and Scopus (Table I).

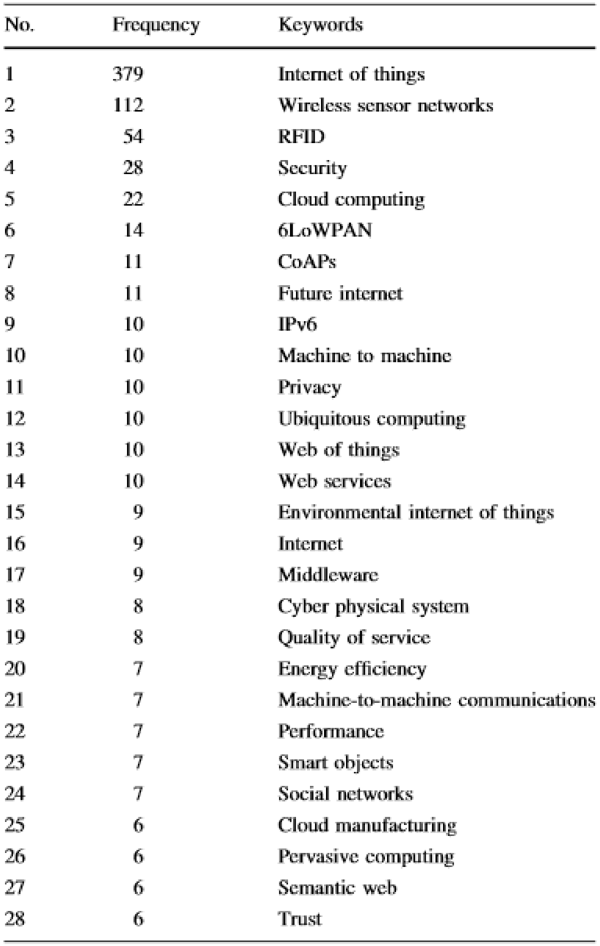
TABLE I. REVISIÓN PRELIMINAR DE ARTÍCULOS EN BASES DE DATOS ESPECIALIZADAS

Keyword of search	Data base	# of results	Reviewed articles	Related articles
Security IOT	Scopus	1.134	89	32
Security internet of things	ACM	224	29	23
Security IOT	IEEE	143	27	21
Total	--	1501	145	76

For this phase a review of the abstracts and conclusions from the preview identified 76 potential publications directly related to the security area in IoT was done. However when checking a large number of related work, it was identified that majority was not related to security issues.

Therefore it held a a second phase where a combination of the greatest number of occurrences of words used in IoT and safety was performed. Study was conducted which in advance by [21] (Table II).

TABLE II. WORD FREQUENCY IN IOT ENVIRONMENTS (YAN, ET AL, 2015)



No.	Frequency	Keywords
1	379	Internet of things
2	112	Wireless sensor networks
3	54	RFID
4	28	Security
5	22	Cloud computing
6	14	6LoWPAN
7	11	CoAPs
8	11	Future internet
9	10	IPv6
10	10	Machine to machine
11	10	Privacy
12	10	Ubiquitous computing
13	10	Web of things
14	10	Web services
15	9	Environmental internet of things
16	9	Internet
17	9	Middleware
18	8	Cyber physical system
19	8	Quality of service
20	7	Energy efficiency
21	7	Machine-to-machine communications
22	7	Performance
23	7	Smart objects
24	7	Social networks
25	6	Cloud manufacturing
26	6	Pervasive computing
27	6	Semantic web
28	6	Trust

For this study purposes, it was taken the fisrt five most frequently phrases, such as: 'IoT and security', 'Middleware', 'RFID', 'Internet', 'Cloud computing', 'Wireless sensor networks' and '6LoWPAN'. To complement this studio, the final third phase was to classify the most frequent problems from the basic security principles. Table III shows the results of this studio.

From the related work summed up in table III, It was identified the two most frequent problems: user authentication, followed by data encryption.

TABLE III
PAPER REVISION IN SPECIALIZED DATA BASES

PROBLEM	# PAPERS	%
User Authentication	33	45,2%
Traffic filter	18	24,6%
Data encryption	25	34,2%
Intrusion detection in real time	1	1,3%
Devices and applications protection	17	23,2%
Secure localization	7	9,5%
Quality service	1	1,3%
Secure connectivity between objects	12	16,4%
Secure protocols	15	20,5%
Information storage	2	2,7%
User resistance	1	1,3%
vulnerable Interfaces	1	1,3%
Cost	3	4,1%
Malware	4	5,4%
Unsecure Software, Hardware	11	15,0%
Unsecure Web interface	2	2,7%
Information theft	9	12,3%

IV. Security model proposed

Due to Internet of Things is a large field with various technologies, a categorization of the issues and technologies was made, this categorization is the basis for analyzing some details of security and privacy in the respective fields.

Figure 2 shows a categorization of the issues and their respective technologies used in each of the topics that make up the Internet of Things.

According with Figure 2, it can be identified eight major areas within IoT which must be specified level of security related studies. They are described detalied below.

- *Communication*: Research on communication protocols has come up with solutions that provide the integrity, authenticity and confidentiality, such as TLS or IPSec. Privacy needs have been addressed by different routing schemes as Onion Routing or Freenet, but these are not widely used.
- *Sensors*: Integrity and authenticity of the sensor data is an objective of the current research that can be handled as watermarking, which was previously described by [22]. The confidentiality of data sensors is a very vulnerable condition; therefore, the need for confidentiality in the sensor is low, so that confidentiality is based on the confidentiality of communication. Mechanisms such as face blurring video data are important to implement in order to preserve the privacy of individuals and objects.

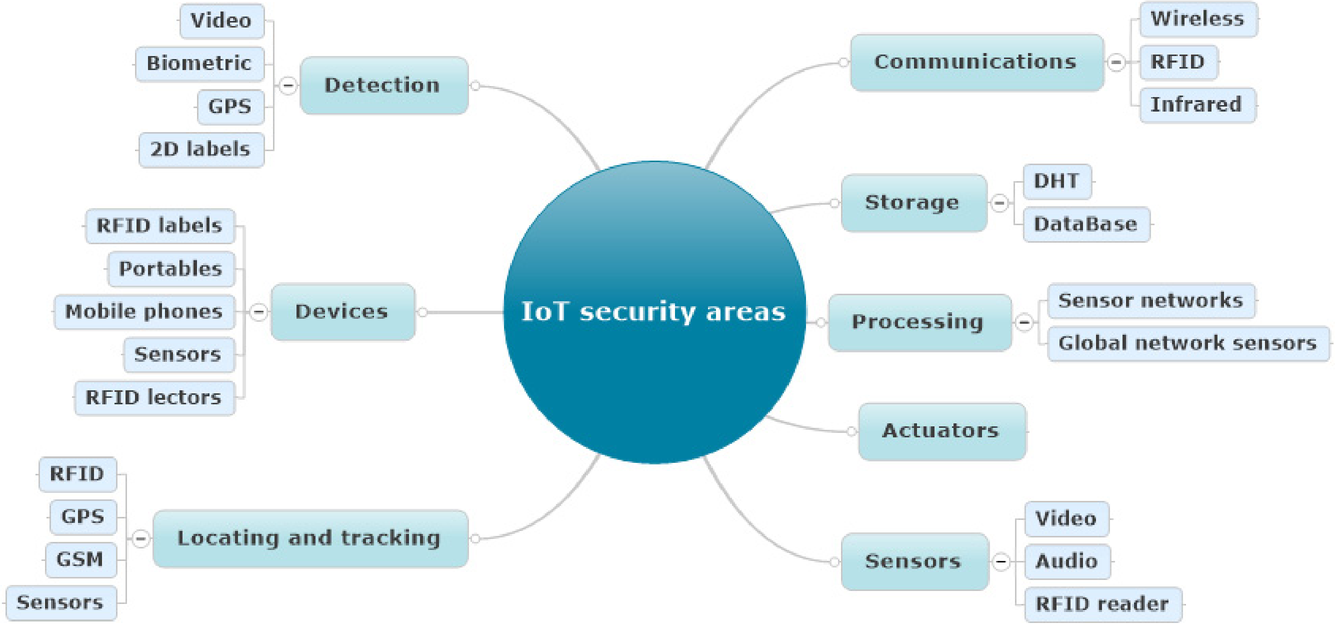


Fig. 2. IoT security areas identified

Sensor availability depends mainly on the communication infrastructure. Regulations are necessary to preserve the privacy of individuals who are currently most often unconscious on the sensors, such as video cameras.

- *Actuator*: Integrity, authenticity and confidentiality of data in an actuator depends primarily on the security of communications.
- *Storage*: Security mechanisms for storage devices are well established. Data storage is highly sensitive to privacy and there are many cases of violation of privacy regulations should be widely distributed to provide an adequate response to user privacy protection. Storage availability depends mainly on the availability of the communication infrastructure and well-established mechanisms for redundancy storage.
- *Devices*: Within the field of integrity of the devices, a device is free from malware. This property has also been called "admissibility" worked by B. Schneier, a presently open issue, researched Trusted Computing Platform (TPM) and highly sensitive. The authenticity of a device handles all the communication parts, not seen such as the end point of connection. Confidentiality is a device with integrity to ensure that no third party has access to internal data devices.

Devices privacy depends on the physical privacy and privacy of communication.

- *Processing*: Integrity in data processing services is based on the integrity of communication devices. Also, it depends on the design and proper execution of algorithms for processing. The authenticity of processing depends solely on the authenticity of the device and the authenticity of the communication.

The property of confidentiality in processing depends only on the integrity of the device, and in the case of distributed processing, depends on the integrity of the communication. The availability of processing depends on the device and the availability of communication exclusively.

- *Location and Tracking*: The integrity of Location and Tracking is based on the integrity of Communication and the integrity of the reference signals used in the location, such as GSM or GPS. It also depends on the authenticity of the authenticity and integrity of communication devices. The confidentiality of data tracking and tracing are of great importance to ensure user privacy and therefore is very sensitive. Confidentiality in this context means that an attacker is not able to disclose the location data and therefore is primarily based on the confidentiality of communication. Data privacy location means that there is no way for an attacker to reveal the identity of the person or object and the location and tracking is not possible without the agreement or explicit knowledge.

TABLE IV

RECOMMENDATION CRITERIA IN SECURITY AREAS

Properties	**Security principles**					
	Integrity	**Authenticity**	**Confidentiality**	**Privacy**	**Availability**	**Regulation**
Communication	High	High	High	Media	High	Low
Sensors	High	Medium	Low	High	Low	High
Actuators	Low	Low	Low	Medium	Low	Media
Storage	High	Medium	High	High	Low	High
Devices	High	Low	Low	Medium	Medium	Medium
Processing	Medium	Low	Low	High	Low	High
Location and tracking	Low	Low	High	High	High	High
Identificación	Media	Baja	Alta	Alta	Alta	Alta

- *Identification:* It uses same sensitivities than Location and Tracking. One difference is the higher sensitivity on the integrity part. It is easier for an attacker to manipulate the identification process as it is handling the localization process. This translates mainly due to technology used (eg RFID or biometrics) is more likely that an attacker manipulate location technologies (eg, GSM).

From this basic classification criteria are defined to determine the relevance of the security level on each of the areas identified in table IV.

V. Conclusions

Since the IOT devices are eminently focused on sending information between devices, or from them to Internet; one of the key measures to be taken, would be the protection of information traveling through them. In most cases this information travels through wireless networks or through public networks, which are vulnerable to being attack.

If communication channel is not adequately protected by encrypting data, it can be easy for an attacker to carry out attacks. The attacker can capture customer traffic, rectify it to pretend to be the originator of it, and send it to the legitimate server, so that it acts as an intermediate point in communications, invisible to both: the source and destination of traffic. Thus, people can get all the information they want even modify it, in order to alter the behavior or performance of the device, or even send false information to users, so they will not take the right decisions regard of the original information.

Another common feature characteristic to a large quantity of IOT devices, is that they use cloud services. In this case these applications have other potential risk; for instance; if there are deficiencies in the management or update the platforms; intruder would be able to access the information store and even take control of the IOT device.

There is a specific need for research into the availability of communication due to DDoS and service provided by IP. In addition, the integrity of the devices must ensure their freedom from malware such as spyware or rootkits, seeing the need for more research. Finally, almost all areas lack mechanisms applicable in the privacy of Internet of Things. The guidelines for Langheinrich are very useful for system designers, but regulations are needed to ensure that systems comply with these guidelines, and mechanisms must be developed to provide users with opportunities to actively protect their privacy rather than relying systems of Internet of Things respect their privacy and implement respective mechanisms.

Finally, it is very well known to use mobile applications that are installed on a Smartphone for any type of management, either obtain data or control the device. As a result, mobile applications can also be the target of attacks, either exploiting vulnerabilities or deficiencies in its implementation, or by developing malicious applications that emulate the behavior and appearance of legitimate access to the IOT devices.

As future work, is foreseen to carry out a characterization of these problems, so that from an ontological model and intelligent agents it can be carried out the appropriate identification of security mechanisms from most frequent problems in clusters of application of IoT. This would facilitate security alternatives identification, deployment access models IoT devices first.

References

[1] D. Boyle, R. Kolcun, and E. Yeatman, "Devices in the internet of things," J. Inst. Telecommun. Prof., vol. 9, no. 4, pp. 26–31, 2015.

[2] D. Pavithra and R. Balakrishnan, "IoT based monitoring and control system for home automation," in 2015 Global Conference on Communication Technologies (GCCT), 2015, pp. 169–173.

[3] V. Vujovic and M. Maksimovic, "Raspberry Pi as a sensor web node for home automation," Comput Electr Eng, vol. 44. pp. 153–171.

[4] R. Aggarwal and M. L. Das, "RFID security in the context of internet of things," in Proceedings of the First International Conference on Security of Internet of Things, 2012, pp. 51-56.

[5] W. Huan, "Studying on Internet of things based on fingerprint identification," in 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), 2010.

[6] R. Weber, "Internet of Things–New security and privacy challenges," Computer Law & Security Review, vol. 26, pp. 23-30, 2010.

[7] L. Atzori, et al., "The internet of things: A survey," Computer networks, vol. 54, pp. 2787-2805, 2010.

[8] G. Gang, et al., "Internet of things security analysis," in Internet Technology and Applications (iTAP), 2011 International Conference on, 2011, pp. 1-4

[9] X. Li, et al., "Research on the architecture of trusted security system based on the Internet of things," in Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on, 2011, pp. 1172-1175.

[10] J. Granjal, et al., "Security for the internet of things: a survey of existing protocols and open research issues," Communications Surveys & Tutorials, IEEE, vol. 17, pp. 1294-1312, 2015.

[11] T. Bécsi, et al., "Security issues and vulnerabilities in connected car systems," in Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on, 2015, pp. 477-482.

[12] M. L. Han, et al., "A Statistical-Based Anomaly Detection Method for Connected Cars in Internet of Things Environment," in Internet of Vehicles-Safe and Intelligent Mobility, ed: Springer, 2015, pp. 89-97.

[13] M. Schellekens, "Car hacking: Navigating the regulatory landscape," Computer Law & Security Review, 2016.

[14] J. Schumann, et al., "R2U2: Monitoring and Diagnosis of Security Threats for Unmanned Aerial Systems," in Runtime Verification, 2015, pp. 233-249.

[15] L. Da Xu, et al., "Internet of things in industries: a survey," Industrial Informatics, IEEE Transactions on, vol. 10, pp. 2233-2243, 2014.

[16] Z. Yan, et al., "A survey on trust management for Internet of Things," Journal of network and computer applications, vol. 42, pp. 120-134, 2014.

[17] D. Meissler, "HP study reveals 70 percent of internet of things devices vulnerable to attack," Retrieved June, vol. 30, p. 2015, 2014.

[18] B. Zhang, et al., "Security architecture on the trusting internet of things," Journal of Electronic Science and Technology, vol. 9, pp. 364-367, 2011.

[19] M. Palattella, et al., "Standardized protocol stack for the internet of (important) things," Communications Surveys & Tutorials, IEEE, vol. 15, pp. 1389-1406, 2013.

[20] R. Piyare, "Internet of things: Ubiquitous home control and monitoring system using Android based smart phone," International Journal of Internet of Things, vol. 2, pp. 5-11, 2013.

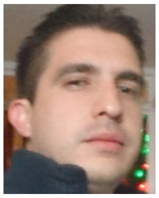
[21] B. Yan, T.-S. Lee, and T.-P. Lee, "Mapping the intellectual structure of the Internet of Things (IoT) field (2000–2014): a co-word analysis," Scientometrics, vol. 105, no. 2, pp. 1285–1300, Sep. 2015.

[22] H. Juma, et al., "On protecting the integrity of sensor data," in Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on, 2008, pp. 902-905.

Paulo Alonso Gaona-García is Associate Professor and active member of GIIRA research group at Engineering Faculty of Universidad Distrital Francisco José de Caldas, Bogotá – Colombia. He obtained his Ph.D. in Information of Engineering and Knowledge at University of Alcalá in 2014. He finished a degree on System Engineer on 2003 and obtained an MSc in Information Science and Communication in 2007 at Universidad Distrital Francisco José de Caldas. His research interest include Web science, network and communications, information security, e-learning, information visualisation and semantic Web.

Carlos Enrique Montenegro Marin is a Ph.D.in systems and computer services for internet from University of Oviedo, Asturias, Spain (2012). He has a Diploma of advanced studies 2008 of the Pontifical University ofSalamanca.He is MSc.Science in Information and Communication Systems from the Universidad Distrital Francisco José de Caldas. He is a System engineer. His research interests include Object-Oriented technology, Language Processors, Modeling Software with, DSL and MDA.

Yuri Vanessa Nieto Acevedo is a MSc. Science Information and Communications from the Universidad Distrital Francisco Jose de Caldas and Industrial Engineer (2012). Is a full time researcher at AXON Investigation Group form CUN University and GIIRA (Investigation Group of Academic Interoperability) member. Her research interests include Learning Analytics, e-Learning, machine learning and virtualization.



Juan David Prieto Rodríguez is Associate Professor at Engineering Faculty of San Mateo Unversity, Bogotá – Colombia. He´s specialist in informatic security at Piloto University. He finished a degree on electronic and telecommunications Engineer on 2009 and obtained graduate postgraduate in 2014. His research interest include, network and communications, information security, information visualisation and digital signal processing.

Securing Cloud Computing from Different Attacks Using Intrusion Detection Systems

Omar Achbarou, My Ahmed El kiram, and Salim El Bouanani

Dept. of Computer Science, Cadi Ayyad Univesity (UCAM), Morocco

***Abstract* — Cloud computing is a new way of integrating a set of old technologies to implement a new paradigm that creates an avenue for users to have access to shared and configurable resources through internet on-demand. This system has many common characteristics with distributed systems, hence, the cloud computing also uses the features of networking. Thus the security is the biggest issue of this system, because the services of cloud computing is based on the sharing. Thus, a cloud computing environment requires some intrusion detection systems (IDSs) for protecting each machine against attacks. The aim of this work is to present a classification of attacks threatening the availability, confidentiality and integrity of cloud resources and services. Furthermore, we provide literature review of attacks related to the identified categories. Additionally, this paper also introduces related intrusion detection models to identify and prevent these types of attacks.**

***Keywords* — Cloud Computing, Cloud Security, Threats, Attacks on Cloud, Intrusion Detection System (IDS)**

I. Introduction

Cloud computing is Internet based infrastructure where shared resources, software and information are provided to computers and other devices on-demand.

The National Institute of Standards and Technology (NIST) defined five characteristics of cloud computing [1]: on-demand self-service, rapid elasticity or expansion, broad network access, resource pooling, and measured service. It also defined three “service models” (software, platform and infrastructure), and four “deployment models” (private, community, public and hybrid) that together categorize ways to deliver cloud services.

Figure 1 shows cloud deployment models together with their internal infrastructure (Infrastructure as a Service IaaS, Platform as a Service PaaS and Software as a Service SaaS), and the essential characteristics of this environment.

Despite the enormous technical and business benefits of cloud computing, concern for security and privacy has been one of the main obstacles that impede its widespread.

In this work, we classify security problems and attacks of cloud computing environments such as Flooding Attack, Denial of Service (DoS) attacks, Side Channel Attacks, phishing, malware Cloud Injection Attacks. To prevent these attackers, Intrusion Detection Systems (IDSs) are effective solutions to resist them. IDS can identify suspicious activities by monitoring network traffic changes, configuration of the system, logs files, and actions of end-users. When such a suspicious event is detected, IDS sends an alert message to a person or monitoring console to trigger some actions for preventing these attacks.

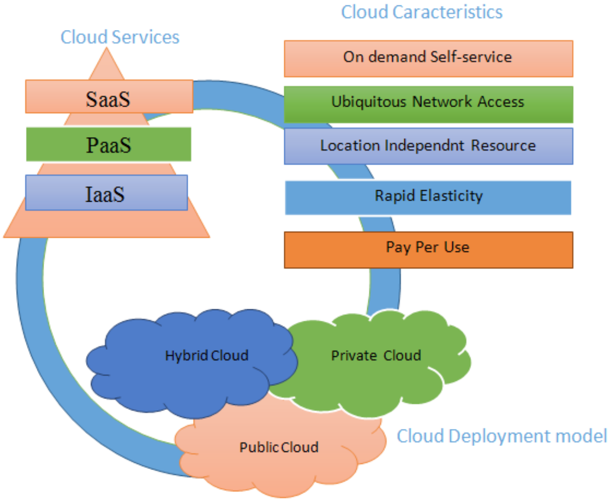


Fig. 1. Cloud deployment models, Characteristics, and infrastructures

The remainder of this paper is structured as follows. The next section presents the main categories of cloud computing security. In Section 3, we present description of the well known attacks affecting cloud computing. Intrusion detection Systems and our types are detailed in section 4. The section 5 presents our proposed model to detect, classify and resist these types of attacks. And the last section summarizes the main contribution of this work and details our perspectives.

II. Categories of Cloud Security

As part of this work, we started an investigation into the security issues and attacks on cloud computing. Cloud computing also suffers from various traditional attacks such as Flooding Attack, Side Channel Attack, port scanning, denial of service (DoS), Distributed Denial of Service (DDoS) etc. We classify these attacks and problems related to the security of cloud computing in five categories, which are summarized in Table 1 [2]:

TABLE I. CLOUD SECURITY CATEGORIES.

Category	Description
Security Standards	Describes the standards required to take precaution measures in cloud computing in order to prevent attacks.
Network	Included network attacks such as denial of service (DoS), DDoS, etc.
Access Control	Included identification, authentication and authorization attacks.
Cloud Infrastructure	Includes attacks each layer of the cloud as SaaS, PaaS and IaaS, it is particularly associated with the virtualization environment.
Data	Covers data related security issues including data migration, integrity, confidentiality, and data warehousing.

DOI: 10.9781/ijimai.2017.439

In addition to identifying cloud security issues and classifying them into several categories, we have identified dependencies among these categories and the security issues they encompass. If one of the categories is prone to certain attacks, other categories may also become prone to these attacks.

III. Attacks Related to the Cloud Security Categories

In what follows, we present a list of attacks on cloud. We briefly explain each attack and accompanied by a brief discussion of the consequences of the attacks in the cloud environment. Table 2 presents a summary of attack names and attack category [2] [7] [9].

A. *Denial of Service Attacks*

A DoS attack is an attempt to make the affected services unavailable to the authorized users In such an attack, the server providing the service is flooded with a large number of applications and therefore the service becomes unavailable for the authorized user. Sometimes when you try to access a website, we see that due to overload, the server with the website is inaccessible and we observe an error message. This occurs when the number of requests that can be processed by a server exceeds its capacity [4].

Thus, the attacker does not have to flood all n servers that provide a certain service in target, but merely can flood a single, Cloud-based address in order to perform a full loss of availability on the intended service [5].

TABLE II. Known Attacks on Clouds.

Attack name	**Category**
Flooding attack	Cloud Infrastructure
Denial of service	Network, cloud Infrastructure
Port Scanning	Network
Attacks on Virtual Machine (VM) or hypervisor	Cloud Infrastructure
Cloud Malware Injection Attack	Cloud Infrastructure, Access
Man-In-The-Middle Cryptographic Attack	Network, Access Control, data
Cross VM side channels	cloud Infrastructure
Phishing	cloud Infrastructure, Network, Access

B. *Port Scanning*

An attack that identifies open, closed and filtered ports on a system in cloud environment [3]. In port scanning, intruders can seize information with the help of open ports like services that run on a system, IP and MAC addresses which belong to a connection, and router, gateway and firewall rules. In the scenario of Cloud, the attacker can attack the services available through the scanning of ports (discovering open ports on which these services are provided) [10].

C. *Malware Injection Attacks*

In the cloud computing, a lot of data is transferred between the cloud provider and the consumer; it is necessary user authentication and authorization [5]. When data is transferred between the cloud provider and the user, the attacker can introduce malicious code between the two actors.

This attack requires the adversary to create its own malicious service implementation module (SaaS or PaaS) or virtual machine instance (IaaS), and add it to the Cloud system. Then, the adversary has to trick the Cloud system so that it treats the new service implementation instance as one of the valid instances for the particular service attacked by the adversary. If this succeeds, the Cloud system automatically redirects valid user requests to the malicious service implementation, and the adversary's code is executed [7].

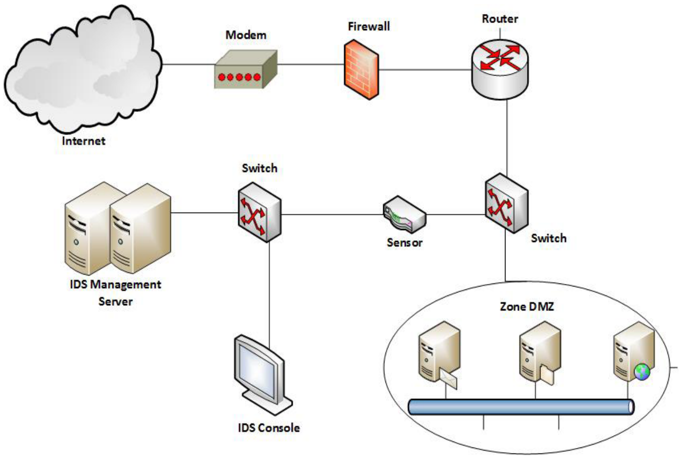


Fig. 2. Network-based Intrusion Detection System architecture

D. *Attacks on Virtual Machine (VM) or hypervisor*

One of the top cloud computing threats involves one of its core enabling technologies: virtualization. In virtual environments, the attacker can take control of virtual machines installed by compromising the lower layer hypervisor. New vulnerabilities, such as zero-day vulnerability found in virtual machines (VM) that attract an attacker access to the hypervisor or other VMs installed. The zero-day vulnerability has been exploited in the application virtualization HyperVM which resulted in the destruction of many websites based on the virtual server [3].

E. *Side Channel Attacks*

These attacks exploit the physical properties of materials to gather information that may give a diagram or pattern of the system to attack. The fact that multiple virtual machines share the same hardware side channel attack makes it relatively easy to achieve. Without implementation of the safety device in the hardware, equipment sharing is dangerous [2].

In cloud computing environments, it is possible to map the infrastructure and identify where the virtual machine resides. It is then possible to instantiate new VMs until one is placed in co-residence with the target VM. After being instantiated, VM attacker can retrieve sensitive data from the legitimate VM attacked. This is a side channel attack-type [16].

F. *Phishing Attacks*

In cloud computing, phishing attacks can be classified into two categories of threats: first, as an abusive behavior in which an attacker hosts a phishing attack site on cloud by using one of the cloud services and second hijack accounts and services in the cloud through traditional social engineering techniques [8].

G. *Man-In-The-Middle Cryptographic Attacks*

This attack is performed when an attacker placed between two users in a cloud environment. Anytime attackers can be placed in the communication path, there is the possibility that they can intercept and change communications [9].

IV. Intrusion Detection System

As detailed in previous section, there are different types of attacks in cloud environment. Intrusion Detection Systems (IDS) are effective solutions to detect and resist these attacks. IDSs are software or hardware systems that realize intrusion detection, log detected information, alert or perform predefined procedures [11, 12].

An IDS is composed of several components [13]:

- Sensors which generate security events.
- Console to monitor events and alerts and control the sensors.
- Central Engine that records events logged by the sensors in a database and uses a system of rules to generate alerts from security events received.

Mainly there are two types of IDS in cloud computing systems: Host based IDS (HIDS) and Network based IDS (NIDS).

A. Host-based Intrusion Detection Systems

A host-based intrusion detection system (HIDS) is a system that monitors a computer system on which it is installed to detect an intrusion and/or misuse, and responds by logging the activity and notifying the designated authority. A HIDS can be thought of as an agent that monitors and analyzes whether anything or anyone, whether internal or external, has circumvented the system's security policy [14].

B. Network based Intrusion Detection Systems

Network-based IDS (NIDS) observe, monitor and analyses the specified and pre-identified network traffic. It can detect different situations based on specified points and generally located between the end point devices like routers, firewalls. A NIDS is an intrusion detection system that attempts to discover unauthorized access to a network by analyzing traffic on the network for signs of malicious activities and events. An example for NIDS architecture and sensor placement is shown in Figure 3 [15].

V. Proposed Work

A. Work

Our proposed model in Figure 3 is a resourceful Cloud IDS which can use a lot of technics to pick up IDS security performance over the Cloud computing. IDS use sensors to check for malicious customer data packets. Initially, the firewall blocks packets from invalid users, otherwise it shipments to the IDS component should analyze them based on predefined rules. The rules are defined based on well known attack strategies by the intruders, it can check the identity of the packets

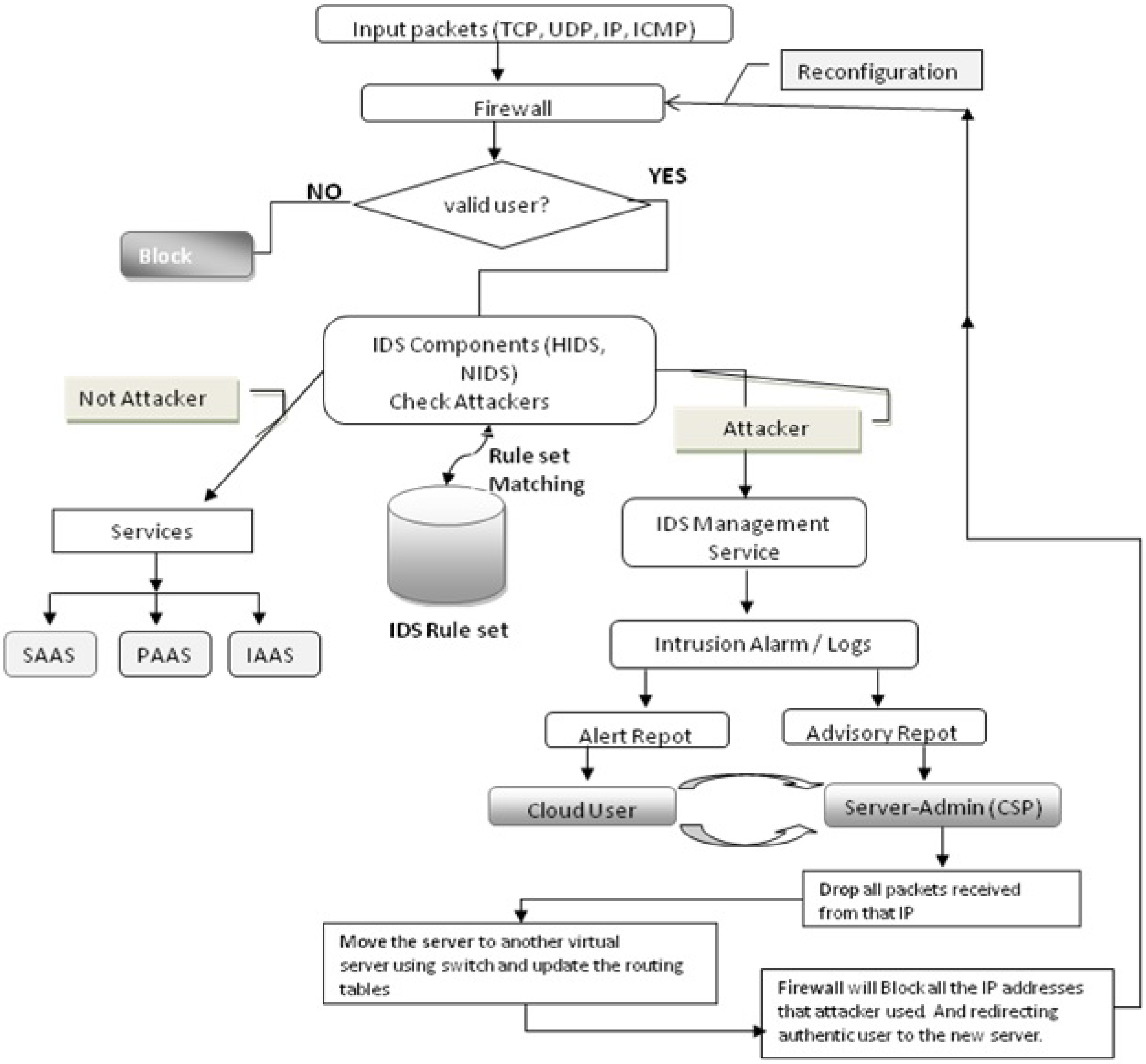

Input packets (TCP, UDP, IP, ICMP)
Reconfiguration
Firewall
NO
YES
valid user?
Block
IDS Components (HIDS, NIDS)
Check Attackers
Not Attacker
Attacker
Rule set
Matching
Services
IDS Rule set
IDS Management Service
SAAS
PAAS
IAAS
Intrusion Alarm / Logs
Alert Repot
Advisory Repot
Cloud User
Server-Admin (CSP)
Drop all packets received from that IP
Move the server to another virtual server using switch and update the routing tables
Firewall will Block all the IP addresses that attacker used. And redirecting authentic user to the new server.

Fig. 3. Proposed IDS Model

if it comes from a pirate, it redirects to the IDS management service that can provide instant reports on the cloud user with an advisory report for the cloud service provider. If it is not from a pirate sends them to the cloud services.

Alerts logs are easily communicated to the user of the Cloud with an expert opinion for the cloud service provider (CSP). The server-admin on examining the security risks involved performs emergency response to the attack by identifying the source IP addresses involved in the attack could automatically generate the access lists that would drop all the packets received from that IP. If the attack type is DDoS attack, the botnet formed by all the zombie machines are blocked. The server-admin then responds to the attack by transferring the targeted applications to virtual machines hosted in another datacenter. Router automation would immediately re-route operational network links to the new location. Hence, the firewall located at the new server will block all the IP addresses that attacker used and if any genuine user is trying to connect to the server, he will be redirected to the new server.

Our model is always dynamic because there are still several days to put launched by this model such as firewall reconfiguration to block new attacker. Also update the IDS database to alert more attacks. So our model is complete to detect such kind of attack.

To manage a large number of data flow packets in such an environment IDS Approach proposed in this paper. IDS able to process huge amount of data and may reduce packet loss. After effective treatment of IDS alerts watched proposed move to a monitoring service by third parties, who in turn informs the cloud directly to users about their system under attack. Figure 3 shows the proposed IDS model. The user cloud access its data on remote servers to the service provider site on the cloud network. Applications and user actions are monitored and recorded by IDS. Alerts logs are easily communicated to the user with a cloud expert advice from cloud service provider.

B. Advantages of proposed model

1. High volume of data in cloud environment could be handled by a single node IDS through a multi-threaded approach.
2. Classify the attack to generate well organized alert Report.
3. Being at a central point, proposed Cloud IDS would be capable to carry out concurrent processing of data analysis, which is an efficient approach.
4. The automatic updates of the routing table on the network to block attacks detected
5. Automatic firewall configuration to block all IP addresses used by the attacker.

VI. Conclusions

Cloud Computing is at the keen interest and numerous works has been published in this field.

This research is primarily done to study the problems and attacks of cloud computing such as DOS Attack, Flooding Attack, and Phishing Attacks on Virtual Machine. Moreover, we classified these attacks into five security categories, namely: security standards, network, access, cloud infrastructure, and data. And we have detailed each one of these attacks. Also this work focuses on the effective solutions to detect this kind of attacks, including intrusion detection systems (IDSs). We propose the deployment of integrated and layered IDS on cloud that designed to cover various attacks.

This IDS integrates knowledge and behavior analysis to increases a cloud‘s security.

References

[1] Final Version of NIST Cloud Computing Definition Published. Available online : http://www.nist.gov/itl/csd/cloud-102511.cfm (accessed on 03 April 2015).

[2] I. Khalil, A. Khreishah, and M. Azeem, "Cloud Computing Security: A Survey," Computers, vol. 3, no. 1, pp. 1–35, Feb. 2014.

[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," J. Netw. Comput. Appl., vol. 36, no. 1, pp. 42–57, 2013.

[4] R. Bhadauria, R. Chaki, N. Chaki, and S. Sanyal, "A Survey on Security Issues in Cloud Computing," Int. J. Innov. Technol. Explor. Eng., vol. 5, no. 6, pp. 83–87, 2011.

[5] M. H. Sqalli, F. Al-Haidari, and K. Salah, "EDoS-shield - A two-steps mitigation technique against EDoS attacks in cloud computing," Proc. - 2011 4th IEEE Int. Conf. Util. Cloud Comput. UCC 2011, pp. 49–56, 2011.

[6] R. Balasubramanian and Dr.M.Aramuthan, "Security Problems and Possible Security Approaches In Cloud Computing," Int. J. Sci. Eng. Res., vol. 3, no. 6, pp. 1–4, 2012.

[7] N. Gruschka and M. Jensen, "Attack surfaces: A taxonomy for attacks on cloud services," Proc. - 2010 IEEE 3rd Int. Conf. Cloud Comput. CLOUD 2010, pp. 276–279, 2010.

[8] M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "On Technical Security Issues in Cloud Computing," in 2009 IEEE International Conference on Cloud Computing, 2009, pp. 109–116.

[9] A. Singh and M. Shrivastava, "Overview of Attacks on Cloud Computing," Int. J. Eng. Innov. Technol., vol. 1, no. 4, pp. 321–323, 2012.

[10] Damien Riquet, Gilles Grimaud and Michaël Hauspie. "Study of the impact of the attacks and distributed multi-path on network security solutions", MajecSTIC, 2012.

[11] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," Natl. Inst. Stand. Technol., vol. 800–94, no. July, p. 111, 2012.

[12] G. Tyler, "Information Assurance Tools Report Intrusion Detection Systems," Information Assurance Technology Analysis Center (IATAC), September 2009.

[13] V. Marinova-Boncheva, "A short survey of intrusion detection systems," Problems of Engineering Cybernetics and Robotics, vol. 58, pp. 23–30, 2007.

[14] K. Vieira, A. Schulter, C. Westphall, and C. M. Westphall, "Intrusion detection for grid and cloud computing," IT Prof., vol. 12, no. 4, pp. 38–43, 2010.

[15] J.-H. Lee, M.-W. Park, J.-H. Eom, and T.-M. Chung, "Multi-level Intrusion Detection System and log management in Cloud Computing," 13th Int. Conf. Adv. Commun. Technol., no. Vmm, pp. 552–555, 2011.

[16] B. Sevak, "Security against Side Channel Attack in Cloud Computing," Int. J. Eng. Adv. Technol., vol. 2, no. 2, pp. 183–186, 2012.

Omar Achbarou is a PhD student, he received his Master's degree in Computer Science from the Cadi Ayyad University Marrakech Morocco. His research interests are computer science, cloud computing security, Internet of Things and big data.

My Ahmed El Kiram is a full professor of computer science at the Department of Computer Science, Faculty of Science Semlalia, Cadi Ayyad University, Morocco. His major field of study is IT security, cryptographic systems and cloud computing. He is the author of numerous publications related to his research interests.

Salim Elbouanani is a PhD student at the Computer Science Department of the Cadi Ayyad University in Marrakesh; Morocco. His is research interests are the security and privacy in the internet of things. He is also working actively in the design of new smart objects solving real problematics in the Marrakesh region.

A MAS-Based Cloud Service Brokering System to Respond Security Needs of Cloud Customers

Jamal TALBI[1], Abdelkrim HAQIQ[1,2]

[1]Computer, Networks, Mobility and Modeling laboratory, Department of Mathematics and Computer, Faculty of Sciences and Techniques, Hassan 1st University, Settat, Morocco,
[2]e-NGN Research group, Africa and Middle East

***Abstract* — Cloud computing is becoming a key factor in computer science and an important technology for many organizations to deliver different types of services. The companies which provide services to customers are called as cloud service providers. The cloud users (CUs) increase and require secure, reliable and trustworthy cloud service providers (CSPs) from the market. So, it's a challenge for a new customer to choose the highly secure provider. This paper presents a cloud service brokering system in order to analyze and rank the secured cloud service provider among the available providers list. This model uses an autonomous and flexible agent in multi-agent system (MASs) that have an intelligent behavior and suitable tools for helping the brokering system to assess the security risks for the group of cloud providers which make decision of the more secured provider and justify the business needs of users in terms of security and reliability.**

***Keywords* — Cloud Computing, Brokering System, Multi-agent System, Security Risk.**

I. Introduction

Cloud computing [1] is a new paradigm of utility computing and enormously growing phenomenon in the present IT industry hype .Many companies, enterprises and organizations outsource some of their information systems to benefit from the cloud services which are Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS). The main interesting features of a cloud are the cost decrease and a faster time to market. Based on sharing resources, the cloud computing changes the user concerns from managing an infrastructure to only focusing on their core business. Currently there are many numbers of providers, but finding the best cloud service provider is difficult. Thus, it is a challenge for the users to choose the more secured cloud provider for fulfilling their requirements.

Nowadays, a few efforts have been devoted to building tools and frameworks that can permit customers to evaluate cloud offerings and rank them based on their ability to meet the user's quality of service (QoS) and security requirements. This is a major problem for every user, especially those who are more concerned about data security and privacy from CSP. For this purpose, cloud brokers [2] have emerged; they can help cloud consumers to select adequate solutions by comparing existing offers, essentially against their prices.

A secure computer system provides guarantees regarding the confidentiality, integrity, availability, non-repudiation and authenticity of its objects (such as data, processes or services). Security is related to vulnerabilities in software, and these are hard to foresee or detect before an actual attack; security involves personal aspects (e.g., user or operator issues) and aspects of the operational environment that are often beyond the control of the development teams. As cloud computing presents new kinds of security risks [3], [4], they need to be treated before wider adoption. Accordingly, we have to dispose a system that measure and rank the secured cloud service providers and then, the cloud services can make a major impact and will craft a healthy competition among cloud providers to satisfy their service level agreement (SLA) and improve their QoS and trustworthiness.

In this paper, our aim is to help a new customer to find the most reliable and secured CP in terms of security and trust through a brokering system integrating multi-agent systems that consists of user agents, providers agents, and broker agents, based on the principle that agent flexibility, intelligence, pro-activity, and autonomy can help cloud computing platforms offer solutions, functionalities, and intelligent services that can define, analyze, measure and rank the cloud service providers using a security risk analysis. Thus, the obtained results make decision of the best option of CP and justify the business needs in terms of security and reliability.

Multi-agent systems [5] represent a distributed computing paradigm based on multiple interacting agents that are capable of intelligent behavior. MASs can often solve problems using a decentralized approach in which several agents cooperate to generate efficient solutions. On the basis of collective AI approaches, developers can embed intelligence within software agents and deploy them on parallel or distributed computers to achieve the high performance required for solving large complex problems while keeping execution time low.

In this context, MASs should include self-detection of failures and self-monitoring of cloud operations and services, QoS security negotiation and SLA management, service-level agreement negotiation [6] [7], cloud interoperability, cloud resource brokering, virtual machines and service migration policies, dynamic scheduling. They're designed to operate in a dynamically changing environment.

The rest of the paper is organized as follows. The next section discusses related work. Section 3 describes the cloud service brokering system. The connection procedure of user and provider agents is presented in Section 4. In Section 5, an implementation and the experiment results in a case study are presented. Finally, Section 6 concludes the paper.

II. Related work

Security metrics are one of criteria that play a major role in ranking service providers. A cloud user may require an efficient, cost effective and basically more secured provider for his application. Since there are many providers who will provide same type of services with different level of security, so it will be a challenge for the user to select. Our motivation in this paper is to promote a novel approach for selecting the secured providers based on measuring security risks of cloud services.

In the same context, many researchers have proposed different approaches to help customer in this mission to select the appropriate cloud service. A collaborative filtering approach [8] rank the items based on similar user's preferences. This algorithm aggregates all the

DOI: 10.9781/ijimai.2017.4310

items purchased by the users and eliminate those items and ask users to rate the remaining services. In [9], cloud rank approach proposed greedy algorithm. It gives a method to rank cloud providers based on existing customer's feedback. It ranks component rather than service of providers. But there is no guarantee that all explicitly rated items by customers are ranked properly. But similar users will experience the same with same cloud providers so for them this approach will be helpful.

QoS-aware web by collaborative filtering [10] proposed a collaborative approach to rank providers on the basis of its web services. This method is useful for the customers who want to get an appropriate cloud provider which provides suitable web services. Thus, this method includes experience of users who used the services already and a hybrid collaborative filtering approach for evaluating web service QoS parameters.

Parveen Dhillon [11] proposed an effective and efficient method to select best cloud service. In order to select the best provider, three parameters are considered. Instead of taking all three parameters together applied. They made a ranking in where the best provider obtained is selected.

Zibin Zheng [12] proposed an approach for ranking equivalent cloud service providers by providing the similar kind of services which will help users to select suitable providers without spending much time for it. This method uses some QoS parameters for predicting best provider.

Deepak Kapgate [13] proposed a predictive broker algorithm based on Weighted Moving Average Forecasting Model (WMAFM). It proposes a new method to balance load on data centers and also minimizes response time. So for end users, they can get their requested service within few seconds.

Subha [14] had done a survey on quality of service ranking cloud computing. Here the author considered few qualities of service parameters and ranked providers based on that.

Cloud Rank [15] approach measures and ranks cloud services for the users. It takes the feedback or rating of users who had used the services already.

An efficient approach [16] find the best cloud provider by using a system for ranking cloud services based on QoS parameters such as service response time, cost, interoperability and suitability. It uses a broker algorithm that classify the existing providers and find out the more effective and efficient provider.

A sophisticated study [17] proposed ranking frameworks in cloud computing based on QoS parameters to select the best possible service provider.

Gani [1] proposed a conceptual model of federated third party cloud ranking and monitoring system (CMFCSPRS) that assures and boosts up the confidence to make a feasible secure and trustworthy market of CSPs.

III. The Conceptual Model of Cloud Service Brokering System (CMCSBS)

We consider the following scenario for explaining our approach. Let a scenario of a new cloud customer; say a company owner or manager is considering adopting cloud facility for the company. Main priority and mandatory condition is to protect company data security and privacy. The manager can see lots of cloud service provider in the market but not adequate guidelines to adopt the best secured cloud service provider for an organization. New cloud customer needs the security and trust certificate or report of these providers for making a decision to choose the right provider in terms of reliability, security and trustworthiness. So, clearly security issues are the most significant issue which is impeding the growth of mobile cloud computing [18] [19]. However, few ranked systems are available in service provisioning or performance issues but not adequate cloud service provider security ranking system is currently available.

In front of the several security issues [20] [21], we need to have some sort of monitoring, assurance and trust which not only come from the cloud service provider but also from a trusted cloud brokering system as shown in Fig. 1.

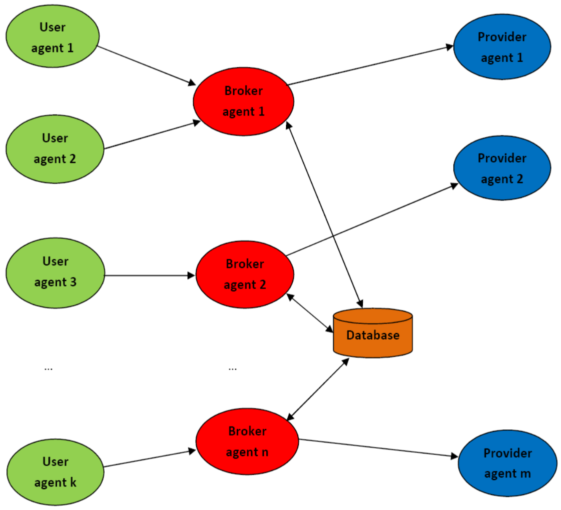


Fig. 1. Overview of the conceptual model of the cloud service brokering system (CMCSBS)

Our cloud service brokering system [22] consists of multiple broker agents, user agents, provider agents, applications and resources. Thus, the proposed model can be described into four-stage in terms of its architecture. First, the user agents send the requests to a broker agent. Second, the broker agent checks whether advertisement and request queues are empty. In case these queues are not empty, the broker agent carries out connection procedures (security needs, risk evaluation and recommendation). In case these queues are overloaded, the broker agent sends those requests (respectively, the advertisements) to other broker agents for balancing the workloads. Third, after executing the connection procedure, the broker agent sends the result to both user and provider agents. Fourth, if a user agent fails to connect to a provider agent, the broker agent recommends another broker agent that has the most potential in the brokering system via the database so that the user agent can send a request to another broker agent. Thereby, the three kinds of agents can be described based on their functionalities as follows.

A. *User Agent*

User agents provide user interface to the users of the system. They post requests to broker agents using message passing. If the connection procedure is completed, they show the results to user through a user interface.

B. *Provider Agent*

Provider agents have similar functionalities as user agents but act on behalf of human providers.

C. *Broker Agent*

The broker agent connects user and provider agents together using the connection algorithm. The broker agent can send a recommendation message based on the historical data of other broker agents in database, so that the broker agent can recommend other broker agents to the user agents which failed to connect to provider agents.

In the summary, the proposed system can act as a middleware between customer and cloud service provider and develops a model to find out the secured cloud service providers based on a connection procedure between the user and provider that will be presented in the next section.

IV. Description of the connection procedure of the CSBS

Probably all cloud service providers have a Service Level Agreements (SLA), but most of these SLAs were written to protect the vendors as opposed to being customer-centric. That has to change, and customers have to demand more with regard to service and the assurance of it. In the same time, cloud providers should protect their data or services from risk and harm. For this aim, the CSBS will conduct vulnerability and threat scans of components and services of the existing providers. The obtained results were fed into the risk evaluation that offer a list ranked of the secured providers.

The connection procedure (security needs, risk evaluation, and recommendation) between users and providers for selecting secured CSPs is presented as shown in Fig. 2. In this context, some assumptions and conditions should be considered as follows [1]:

- The CSBS must maintain the trust and reliability.
- The CSBS has enough resources to provide for processing and executing their own work.
- The system must be maintained and regulated by strict laws and transparent policies.
- Both the CSBS and CSPs mutually agree before executing the software penetration test.
- We consider that a CSP provide IaaS, PaaS and SaaS of its own.
- The CSBS is only the responsible of computing security metrics from sources and processes these measures for ranking results.
- A new cloud user looking for security and reliability should pay to the CSBS to see the ranked results.

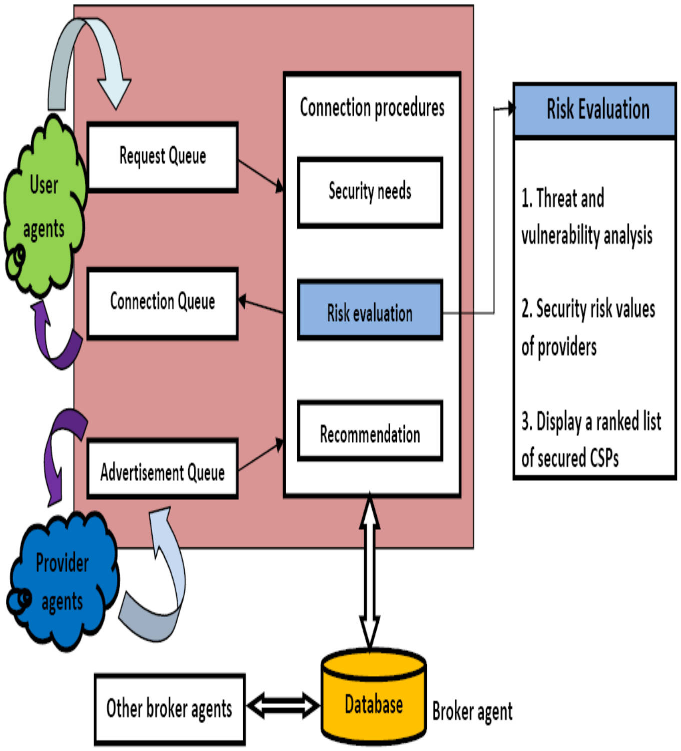


Fig. 2. The architecture of the CSBS

A. Security Needs Stage

The broker collects security requirements from user. It may be infrastructure requirements, platform requirements or software requirements. It uses the five CIANA objectives (Confidentiality, Integrity, Availability, Non-Repudiation, and Authenticity) to define the security need of each cloud user. If the customer needs the objective, the value is equal to 1, otherwise to 0.

B. Risk Evaluation Stage

All the registered cloud service providers give all the services which they are providing. Cloud broker contains the level of security of cloud providers. So the client gives requirements to broker, it checks the provider's performance based on criteria that are risks computed.

1) Threat and Vulnerability Analysis:

A vulnerability is a software defect or weakness in the security system which might be exploited by a malicious user causing loss or harm [23]. The identification of these vulnerabilities has been used by several approaches and researchers to estimate risks of the systems. In our case, we take into account five cloud security threats given by the Cloud Security Alliance (CSA) [24] to evaluate the risks. These threats are each related to the 5 CIANA objectives:

- Data Breaches = {Confidentiality}
- Data Loss = {Availability, Non-Repudiation}
- Account Hijacking = {Confidentiality, Integrity, Availability, Non-Repudiation, Authenticity}
- Insecure Interfaces = {Confidentiality, Integrity, Authenticity}
- Denial of Service = {Availability}

We combine these relations with the security needs of each cloud user to obtain a function called harm. This later is defined on each customer, for each threat through the sum of the affected security needs. For example, the Insecure Interfaces threat (t) has the following harm on the cloud user (k) with the security needs (Confidentiality, Integrity, Non-Repudiation):

$$Harm\ (t,\ k) = (1\times 1) + (1\times 1) + (0\times 0) + (0\times 1) + (1\times 0) = 2 \quad (1)$$

where the first value of each bracket is equal to 1 if the threat corresponds to the objective, 0 otherwise, and the second value is related to the security need.

2) Measuring Security Risk Assessment:

Once we calculate the harm of the threats on each cloud user, we have to determine the response to these threats for each cloud provider. For this aim, we use the STAR Registry and the matrix defined by the CSA [24].

The CSA matrix defines a list of security controls that a cloud provider should implement to reduce security risks. Each of these controls can be related to one or multiple threats. In addition, the STAR Registry publishes the list of implemented controls for providers willing to follow these recommendations.

In our case, we use these two information as binary values (a control mitigates a threat or not / a control is implemented by a provider or not) to calculate the coverage score, which indicates the response of a provider to a given threat. This value is a percentage, if the provider implements all controls mitigating a threat, it gets a coverage for this threat of 100%. In our case, this percentage is brought to a score on a scale of 0 to 5 (with 5 equivalent to 100%).

Usually, the vulnerability is assessed and used to calculate a risk value of an information system [2]. But in a cloud context, providers may be tempted to conceal their vulnerabilities for security reasons. This is why we use the coverage based on the security controls. By using the maximum possible coverage value Covgmax (in our case 5), it is possible to get an equivalent to the vulnerabilities. Therefore, by combining this value with the harm we can define the following risk formula for a threat t, a cloud user k and a provider CSP p:

$$Risk(t,k,p) = Harm(t,k) + (Covg_{\max} - Covg(p,t)) \quad (2)$$

3) List Ranked of the Secured CSPs:

The CSBS model provides optimal cloud service provider selection from the more numbers of CSPs based on security risk values estimated in the last step which provides a list ranked of the more secured CSPs for each customer want to see the ranked results.

C. Recommendation Stage

After the risk evaluation stage, some of the user requests may fail to be matched to the appropriate provider. This failure likely originated in the fact that the users' requests and their matching providers are processed by different broker agents. In this case, a heuristic strategy is applied to seek another broker agent that has the most potential in brokering. The user agent will connect with this broker and start a new cycle.

A database is designed to handle the recommendation requests from broker agents. The broker agent attempts to make a suggestion by predicting the current advertised information of the provider agents based on their historical data in database. To implement this strategy, the broker agents periodically update the information about all of the provider agents connecting to it. The historical data represents the statistical pattern and provides the predictive information to the broker agent. The steps of recommendation are described as follows:

- After risk evaluation stage, broker agent 1 makes a list of requests of user agents connecting to it that failed to be matched.
- Broker agent 1 accesses the database to obtain a suggestion for the potential broker agent for each request.
- With each request, the broker agent looks into the risk value of providers to recommend another broker agent 2 that has the lowest risk value. The information about broker agent 2 will be sent back to the user agent by broker agent 1.
- The user agent will connect to broker 2 and starts a new cycle.

V. Implementation and Experiments Results

To demonstrate the feasibility and the efficiency of our approach, we illustrate a series of simulations using the architecture of the CSBS described in Section IV in case study with four cloud users CU 1, CU 2, CU 3 and CU 4 under some threats related to the CIANA objectives requesting services from five cloud providers X, Y, Z, T and W.

The security requirements step provides the needs of our customers using the user agents in terms of CIANA objectives (see Table 1). Then, the harm function on each cloud customer will be computed (see Table 2) and added to the coverage of the cloud providers for the 5 cloud threats (see Table 3) to obtain the maximum risk values corresponding to our cloud users for each provider by exploiting our CSBS functionalities in this case study.

TABLE I. SECURITY NEEDS OF THE FOUR CLOUD USERS

	Confidentiality	Integrity	Availability	Non-Repudiation	Authenticity
CU 1	1	1	0	1	0
CU 2	0	1	1	1	1
CU 3	1	0	1	0	0
CU 4	1	0	0	1	1

TABLE II

CALCULATION OF THE HARM VALUES ON EACH CLOUD USER

	CU 1	CU 2	CU 3	CU 4
Data Breaches	1	0	1	1
Data Loss	1	2	1	1
Account Hijacking	3	4	2	3
Insecure Interfaces	2	2	1	2
Denial of Service	0	1	1	0

TABLE III

COVERAGE OF THE CLOUD PROVIDERS FOR THE 5 CLOUD THREATS

	CSP X	CSP Y	CSP Z	CSP T	CSP W
Data Breaches	3	5	4	1	2
Data Loss	5	3	4	4	2
Account Hijacking	1	4	3	2	5
Insecure Interfaces	2	5	5	1	3
Denial of Service	3	1	4	1	4

TABLE IV

MAXIMUM RISK VALUES OF THE CUS FOR EACH PROVIDER

	CSP X	CSP Y	CSP Z	CSP T	CSP W
CU 1	7	4	5	6	4
CU 2	8	5	6	5	7
CU 3	6	5	4	5	4
CU 4	7	4	5	6	4

Fig. 3 shows the comparison between the risks in cloud customers for the five cloud providers by using the broker agents presenting in our CSBS. Thus, the user can request services by starting with the providers having the minimum security risks [16].

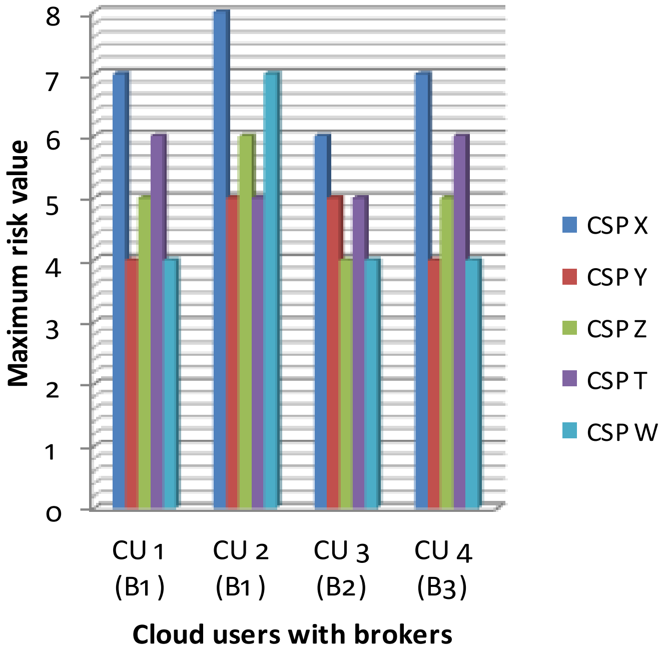


Fig. 3. Comparison of risks in cloud users for the five cloud providers

VI. Conclusion and Future Work

In this paper we have presented a MAS- based cloud service brokering system to respond the security needs of the cloud customers in the aim to deliver different types of services. So, the multiple cloud service providers make a dilemma for a cloud user to choose each provider is more secured and has the minimum security risk. Hence, we propose a cybersecurity model based on three stages used in the connection procedure of the cloud service brokering system. In this work, broker agents are introduced to make our approach more flexible and efficient which can handle a huge amount of user requests by implementing this system in a case study and comparing the empirical results. In the future, we plan to continue the current research work to allow CSBS to be extended in a real use cases, then combining the risk values with costs to make decisions for the cloud provider selection.

References

[1] M. Whaiduzzaman and A. Gani, "Measuring Security for Cloud Service Provider: A Third Party Approach", International Conference on Electrical Information and Communication Technology (EICT), pp. 1-6, 2013 IEEE.

[2] G. Elio, K. Dahman, and B. Gateau, C. Godart, "A Broker Framework for Secure and Cost-Effective Business Process Deployment on Multiple Clouds", CAiSE 2014 Forum/Doctoral Consortium, Thessaloniki, Greece. June 2014.

[3] Cloud Security Alliance. Cloud Control Matrix/Security, Trust & Assurance Registry/Consensus Assessments Initiative Questionnaire. Technical report.

[4] European Network and Information Security Agency. Benefits, risks and recommendations for information security. Technical report, 2009.

[5] D. Talia, "Clouds Meet Agents: Towards Intelligent Cloud Services", IEEE Internet Computing, Vol.16, pp. 78-81, 2012.

[6] J. Yan, R. Kowalczyk, J. Lin, M.B. Chhetri, S.K. Goh, and J. Zhang, "Autonomous Service Level Agreement Negotiation for Service Composition Provision", Future Generation Computer Systems, Vol. 23, No. 6, pp. 748-759, 2007.

[7] K.M. Sim, "Agent-based Cloud Computing", Services Computing, IEEE Transaction on, Vol. 5, No. 4, pp. 564-577, 2012.

[8] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering", IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan. /Feb. 2003.

[9] Z. Zibin, Z. Yilei, and M. R. Lyu, "Cloud Rank: A QoS-Driven Component Ranking Framework for Cloud Computing" in Reliable Distributed Systems, 29th IEEE Symposium on 2010, pp. 184-193.

[10] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS- Aware Web Service Recommendation by Collaborative Filtering", IEEE Trans. Service Computing, vol. 4, no. 2, pp. 140-152, Apr.-June 2011.

[11] P. Dhillon and V. Arora, "A Compositional Approach of Reliable and Efficient Cloud Service Selection", Volume 2, Issue 8, August 2012 ISSN: 2277 128X, International Journal of Advanced Research in Computer Science and Software Engineering.

[12] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang, "QoS Ranking Prediction for Cloud Services", Parallel and Distributed Systems, IEEE Transactions on, vol.24, no. 6,pp. 1213-1222,June 2013.

[13] D. Kapgate, "Weighted Moving Average Forecast Model based Prediction for Service Broker Algorithm for Cloud Computing", International Journal of Computer Science and Mobile Computing, vol. 3, Issue. 2, February 2014.

[14] M. Subha and M. U. Banu, "A Survey on QoS Ranking in Cloud Computing", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 2, February 2014.

[15] R. Yuvarani and M. Sivalakshmi, "Achieve Ranking Accuracy Using Cloud Rank Framework for Cloud Services", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Special Issue 1, March 2014.

[16] K. Amrutha and B. Madhu, "An Efficient Approach to Find Best Cloud Provider Using Broker", International Journal of Advanced Research in Computer Science and Software Engineering 4(7), pp. 943-946, July 2014.

[17] P. Bathla and S. Vashit, "A Sophisticated Study of QoS Ranking Frameworks in Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.4, Issue 7, July 2014.

[18] M.T. Khorshed, A.B.M.S. Ali, and S.A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing", Future Generation Computer Systems, vol. 28, pp. 833-851, 6//2012.

[19] R. Buyya, Y. Chee Shin, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," in High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on, 2008, pp. 5-13.

[20] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Systems, vol. 25, pp. 599-616, 6//2009.

[21] S.M. Habib, S. Ries, and M. Muhlhauser, "Towards a Trust Management System for the Cloud Computing," in Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on, 2011, pp. 933-939.

[22] J. Kang and K. M. Sim, "Towards Agents and Ontology for Cloud Service Discovery," 2011 IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.

[23] C.P. Pfleegerc and S.L. Pfleeger, Security in Computing, 3rd edition, Prentice Hall, 2003.

[24] Cloud Security Alliance. Cloud Control Matrix / Security, Trust & Assurance Registry/Consensus Assessments Initiative Questionnaire. Technical report.

Jamal TALBI received the B.Sc. in Computer Sciences from the University of Hassan 1st, Faculty of Sciences and Techniques (FSTS), Settat, Morocco, in 2009, and M.Sc. degree in Business Intelligence from the Sultan Moulay Slimane University, Faculty of Sciences and Techniques (FSTBM), Beni Mellal, Morocco, in 2011. Currently, he is working toward his Ph.D. at FSTS. His current research interests include decision support, information systems, networking architectures, security, privacy, cryptography in cloud computing.

Abdelkrim HAQIQ has a High Study Degree (DES) and a PhD (Doctorat d'Etat), both in the field of modeling and performance evaluation of computer communication networks, from the University of Mohamed V, Agdal, Faculty of Sciences, Rabat, Morocco. Since September 1995 he has been working as a Professor at the department of Mathematics and Computer at the Faculty of Sciences and Techniques, Settat, Morocco. He is the Director of Computer, Networks, Mobility and Modeling laboratory and the responsible for engineering education in Computer Engineering at the same Faculty. He is also the General Secretary of the electronic Next Generation Networks (e-NGN) Research Group, Moroccan section. Dr. Abdelkrim HAQIQ is actually Co-Director of a NATO multi-year project and Co-Director of a Moroccan Tunisian research project. Dr. Abdelkrim HAQIQ's interests lie in the areas of modeling and performance evaluation of communication networks, cloud computing and security. He is the author and co-author of more than 80 papers (international journals and conferences/workshops). He was a publication co-chair of the fifth international conference on Next Generation Networks and Services, held in Casablanca, May, 28 - 30, 2014. He was also an International Steering Committee Chair and TPC Chair of the international conference on Engineering Education and Research 2013, iCEER2013, held in Marrakesh, July, 1st –5th, 2013, and a TPC co-chair of the fourth international conference on Next Generation Networks and Services, held in Portugal, December, 2 - 4, 2012. Dr. Abdelkrim HAQIQ was the Chair of the second international conference on Next Generation Networks and Services, held in Marrakech, July, 8- 10, 2010. He is also a TPC member and a reviewer for many international conferences. He was also a Guest Editor of a special issue on Next Generation Networks and Services of the International Journal of Mobile Computing and Multimedia Communications (IJMCMC), July-September 2012, Vol. 4, No. 3, and a special issue of the Journal of Mobile Multimedia (JMM), Vol. 9, No.3&4, 2014.

Use Trust Management Framework to Achieve Effective Security Mechanisms in Cloud Environment

Hicham Toumi[1], Bouchra Marzak[1], Amal Talea[2], Ahmed Eddaoui[2], Mohamed Talea[1]

[1]*Information Processing Laboratory, Department of Physical, University Hassan II Casablanca, Morocco*
[2]*Department of Mathematics and Computer Science, University Hassan II, Casablanca, Morocco*

***Abstract* — Cloud Computing is an Internet based Computing where virtual shared servers provide software, infrastructure, platform and other resources to the customer on pay-as-you-use basis. Cloud Computing is increasingly becoming popular as many enterprise applications and data are moving into cloud platforms. However, with the enormous use of Cloud, the probability of occurring intrusion also increases. There is a major need of bringing security, transparency and reliability in cloud model for client satisfaction. One of the security issues is how to reduce the impact of any type of intrusion in this environment. To address this issue, a security solution is proposed in this paper. We provide a collaborative framework between our Hybrid Intrusion Detection System (Hy-IDS) based on Mobile Agents and virtual firewalls. Therefore, our hybrid intrusion detection system consists of three types of IDS namely IDS-C, IDS-Cr and IDS-M, which are dispatched over three layer of cloud computing. In the first layer, we use IDS-C over our framework to collect, analyze and detect malicious data using Mobile Agents. In case of attack, we collect at the level of the second layer all the malicious data detected in the first layer for the generation of new signatures using IDS-Cr, which is based on a Signature Generation Algorithm (SGA) and network intrusion detection system (NIDS). Finally, through an IDS-M placed in the third layer, the new signatures will be used to update the database NIDS belonging to IDS-Cr, then the database to NIDS belonging of IDS-Cr the cluster neighboring and also their IDS-C. Hardware firewall is unable to control communication between virtual machines on the same hypervisor. Moreover, they are blind to virtual traffic. Mostly, they are deployed at Virtual Machine Monitor- level (VMM) under Cloud provider's control. Equally, the mobile agents play an important role in this collaboration. They are used in our framework for investigation of hosts, transfer data malicious and transfer update of a database of neighboring IDS in the cloud. With this technique, the neighboring IDS will use these new signatures to protect their area of control against the same type of attack. By this type of close-loop control, the collaborative network security management framework can identify and address new distributed attacks more quickly and effectively.**

***Keywords* — Cloud Computing, Virtual Firewalls, Mobile Agents, Security.**

I. Introduction

CLOUD computing represents a distributing computing mechanism that by the use of the high speed network and highly scalable distributed computing platforms in which computational resources are offered 'as a service'. Cloud computing architecture introduces many technologies including server virtualization, Network Virtualization (NV), and Network Function Virtualization (NFV) to enhance the essential characteristics of cloud computing [1][2]. Cloud services allow individuals and enterprises to use software and hardware that are managed by providers at remote locations. It is a model for enabling scalable, on demand network access to a shared pool of configurable computing resources that can be provisioned ubiquitously and released with minimal management effort and cloud service provider interaction [3][4]. At the same time, the transformational nature of the cloud is associated with significant security and privacy risks [5][17].

Therefore, the intrusion detection or confidentiality of data over Cloud is one of the glaring security concerns. The fast growth of cloud computing technology introduces more of the vulnerabilities. Security is considered to be one of the most critical aspects in cloud computing environment due to the confidential and important information stored in the cloud [5][6]. Network security appliances, such as Intrusion Detection Systems (IDS) is widely deployed in advantage points and play an important role in protecting the network from attacks. That is why; it is nowadays widely deployed for securing critical IT-Infrastructures. Due to different deployment mechanisms, we can distinguish different types of IDS; IDS can be categorized as software-based IDS, hardware-based IDS, and VM-based IDS [7]. Most of these appliances work without collaboration, their detection results are isolated and cannot be collected and analyzed systematically. Therefore, we thought of a new security policy that allows the detection of distributed attacks such as deny of service (DoS) and Distributed Denial of Service (DDoS) [6].

In this paper, we will deepen the development of our approach based in principle on the cooperation of the Hybrid Intrusion Detection System (Hy-IDS), Firewall and mobile agents. The cooperation between Hy-IDS, Firewall and mobile agents present what is called a Framework. This framework allows to reach four objectives: the first, detection intrusion in a virtual environment using mobile agents for collecting malicious data. The second, generating new signatures from malicious data, which were collected in the first phase. The third, dynamic deployment of remote response actions using virtual firewall. Finally, dynamic deployment of updates between clusters in a cloud computing, using the newest signatures previously created.

The rest of this paper is organized as follows: The section II presents theoretical background and discusses some related works in the area of Mobile Agent-based IDS and NIDS. The section III forms the core of this paper explains and describes in detail our approach. Whereas the proposed framework is discussed in section IV. Finally, we give conclusion, perspective and references in section V.

II. Theoretical Background and Related Work

In this section, we start with theoretical background include cloud computing, mobile agent technology in cloud computing and Signature Generation Algorithm as the first part, and Related Work as a second part.

DOI: 10.9781/ijimai.2017.4311

A. Cloud Computing

Cloud computing allows accessing resources and services offered by servers from different places. Therefore, it is a model of distributed computing [5] [8][9]. It is undergoing an incontestable success, which could be indeed compromised by concerns about the risks related to potential misuse of this model aimed at conducting illegal activities. To provide secure and reliable services in cloud computing environment is an important issue. Then, there have been a great deal of inherent issues in cloud computing such as data security, vulnerability management, disaster recovery system, and business continuity process and identity management [10]. Then, there are numerous security issues in cloud computing as it encompasses many technologies including networks, virtualization, load balancing, operating systems, transaction management, resource scheduling, concurrency control and memory management [11]. Virtualization enables customers to run multiple operating systems concurrently on a single physical server, where each of the operating systems runs as a self-contained computer [12][13]. More recently, virtualization at all levels became important again as a way to improve system security, reliability and availability, reduce costs, and provide greater flexibility.

B. Mobile Agent Technology in Cloud Computing

The Mobile Agent has its applications in many areas including network management, mobile computing, information monitoring, searching information, remote software management and others. Mobile Agents enhance the performance in these areas by providing the following services [14][6]: there are efficiency and reduction of network traffic, interaction with real-time entities, life cycle of mobile agent and convenient development paradigm.

C. Signature Generation Algorithm (SGA)

Different sessions of attacks are given as input to Signature Generation Algorithm (e.g, Apriori Algorithm and Signature Apriori Algorithm). According to support and confidence value rule are generated by Signature Generation Algorithm. These rules are given to IDS. When attack is generated for which signature is stored in IDS, it generates alarm [5].

D. Relevant Works and Limitations

In the literature there are few works that use IDS, NIDS (Snort and signature apriori algorithm) and mobile agents in the cloud computing.

Chirag N. Modi et al propose a framework integrating network intrusion detection system (NIDS) in the Cloud. Then, NIDS module consists of Snort and Signature Apriori Algorithm. It generates new rules from captured packets. These new rules are appended in the Snort configuration file to improve efficiency of Snort. The objective of this approach is to reduce impact of network attacks (known attacks as well as derivative of known attacks). Derivative attacks can be detected by Snort [15]. However, this work is unable to detect intrusion at the hosts, and Distributed denial of service attacks (DDOS) [6].

In [16] the VMs are attached to MA, which collects evidences of an attack from all the attacked VMs for further analysis and auditing. Then, they have to correlate and aggregate that data to detect distributed attacks. This work tried to offer a line of defense by applying mobile agent's technology to provide intrusion detection for cloud applications regardless of their locations. Thus, it builds up a robust distributed hybrid model scalable, flexible and cost effective method based on mobile agents (MA).

After that, we found the need for collaboration between several security solutions. This collaboration is mainly based on mobile agents. Then we exploit mobile agents for security against intrusion attacks and at the same time as a communication tool between different layers of cloud computing.

III. Our Framework for Trust Management in Cloud Environments

The designed dynamic network security architecture for IaaS platforms is based on the mechanisms of VM traffic redirection and policy management, security-supporting services. Our previous works [5][6]. Today, we present a new approach based on the improvement of collaboration among Hybrid Intrusion Detection System (Hy-IDS), Signature Generation Algorithm (SGA), Mobile Agents (MA) and Firewall. It follows the principle the P2DR (Policy, Protection, Detection, and Response).

A. Challenges of the Framework Proposed

The objectives of our framework are grouped into four main Points as follows:

1. Intrusions detection in a virtual environment using mobile agents in order to collect malicious data.
2. Generating new signatures from malicious data, which were collected in the first part.
3. Dynamic deployment of updates between clusters in a cloud computing, using the newest signatures previously created.
4. Dynamic deployment of remote response actions using virtual firewall.
5. Dynamic deployment of updates between clusters in a cloud computing, using the newest appropriate response actions previously created.

B. Our Proposed Hybrid Framework and Cloud Computing

1) Components of our framework

Our framework based on many concepts and components as follows: Hybrid Intrusion Detection System (Hy-IDS) combines Intrusion Detection System Center (IDS-Cr), Intrusion Detection System Control (IDS-C) and Intrusion Detection System Master (IDS-M). The IDS-Cr based on an Intrusion Detection System (IDS) and Signature Generation Algorithm (SGA). However, IDS-C is based on the combination of IDS with the living environment of mobile agents named Agents Agency (AA). The IDS-M is based on Intrusion Detection System (IDS) and Living Environment of Mobile Agents named Agents Agency (AA). Concerning the types of IDS; there are network based (NIDS) and host based (HIDS) intrusion detection systems. Then, some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Finally, using mobile agents to ensure communication between the IDS-C, IDS-Cr and IDS-M [6].

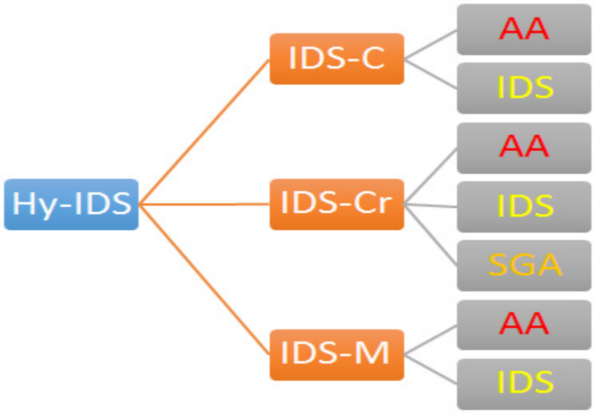


Fig. 1. Components of our Hy-IDS.

2) *Proposed framework over cloud computing*

Cloud architecture used with our framework is presented as a front-end and back-end. Front-end is connected to both external network as well as internal network. Then, It is presented in the figure 2 by the Cloud-layer include only the Cloud Controller (CLC). It is used by the user for communicate with Cloud Computing. It allows management of cloud security. However, back-end consists of computer hardware and software (servers, storage), that are designed for the delivery of services. The CLC acts as the administrative interface for cloud management and performs high-level resource scheduling, and handles reporting, authentication and accounting. The Cluster Controller (CC) acts as the front-end for a cluster within a cloud computing and communicates with the Cloud Controller and Node Controller. Finally, the Node Controller (NC) at level of physical server; it hosts the virtual machine instances and manages the virtual network endpoints.

We have just presented the cloud architecture and the components of our Hy-IDS. Therefore, we proceed to the establishment or distribution of the components of our framework on this architecture according to our strategy the protection. However, Our Hybrid Intrusion Detection System (Hy-IDS) combines Intrusion Detection System Control (IDS-C) and Intrusion Detection System Center (IDS-Cr), which are placed in the back-end. Finally, Intrusion Detection System Master (IDS-M), which is placed in the front-end. Then, the general architecture of our framework, shown in Figure 2, is divided into four main layers interact.

IDS-C: VMs are further managed by hypervisors, also known as Virtual Machine Monitor (VMM) and are basically installed on server hardware. Thus, we use VMM in our framework to ensure a new level of trust in the VMs. Then, we place the components of IDS-C at the level of nodes (physical server) for monitoring virtual machines. For more details, we place IDS-C at the level of VMM. At the same time, we place specific static agent detectors (SA) at the level of VMs. Our IDS-C is based on the cooperation of IDS with the living environment of mobile agents named Agents Agency (AA).

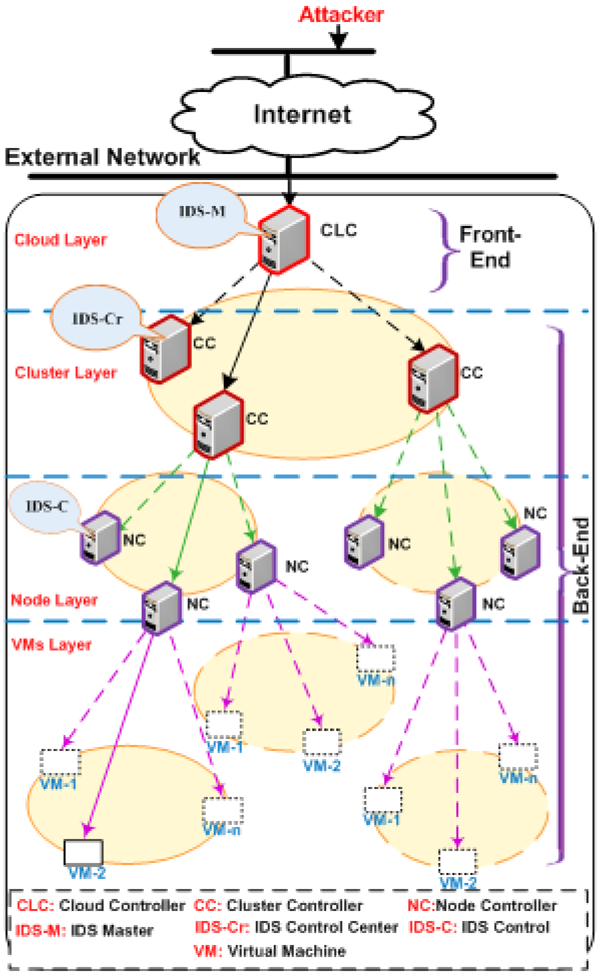


Fig. 2. The Hierarchy of our cloud computing.

IDS-Cr: it installed in the front-end Cluster for the monitoring of nodes. It also generates new signatures. It consists of an Intrusion Detection System (IDS) and Signature Generation Algorithm (SGA).

IDS-M: it is placed in the front-end Cloud for the monitoring of Clusters and Management of Update (new signatures). The IDS-M is based on IDS and Living Environment of Mobile Agents named Agents Agency (AA). Finally, all communication between these components is provided by mobile agents.

3) *Intrusion detection management based on our Hy-IDS*

VM-layer and Node-layer constitute the fundamental design of our proposed framework. Then, each node consists of three main components namely IDS Control (IDS-C), Agents Agency (living environment of mobile agents), Specific Static Agent Detectors (SA) [6].

Static Agents (SA) placed at the level of virtual machines. It generates an alert whenever they detect suspicious activities, then send alert's ID to IDS-C. In this case, IDS-C will send investigative Mobile Agent (IMA) with a specific task, to each agency (VM) that sent similar alerts. The IMA visit and investigate all those VMs for collecting information, who affirm the existence of an intrusion. It carries back the result at to the IDS Control to perform advanced analysis.

In case of attack, IDS-C aggregate malicious data, then placing them in a temporary database. After, IDS-C uses Transfer Mobile Agents (TMA) for notifying IDS-Cr placed in the cluster layer as shown in the figure 3 [6]. After, IDS-Cr dispatches Investigative Mobile Agents (IMA) to any IDS-C those send TMA, for aggregation and collection of their malicious data from the database temporarily. Then, IDS-Cr uses all malicious data collected by IMA and using them to generate new signatures through a Signature Generation Algorithm (SGA) at level of IDS-Cr.

Finally, these new signatures will be used to update the database IDS belonging to this IDS-Cr. after that, IDS-Cr sends these new signatures toward IDS-M. Thus, IDS-M uses these new signatures to update databases of neighboring cluster (eg: IDS in CC_2 and CC_3) based on update mobile agents (UMA) as shown in figure 3.

These updates go through the IDS-M, to maintain a hierarchical structure in our framework. Then, our framework protects neighboring clusters of the same type of attack. Thus, among the advantages of our approach, other clusters are protected against the same category attack.

C. *Responses to Attacks Using Virtual Firewall*

The essential role, which can perform a firewall is that of securing the connections between the tenant network and the cloud infrastructure network. The cloud infrastructure network really hosts a number of traffic profiles, such as management traffic, storage traffic, and management traffic, Cluster/CSV traffic and Live Migration traffic. However, these traffic profiles must be totally analyzed and controlled using firewall. In our approach, we use the firewall to respond to incoming and outgoing attacks in a hypervisor. Consequently, the virtual firewall is a network security system, which controls incoming and outgoing network traffic based on a set of rules.

However, the firewall could be used to control access between virtual machines and Internet access. Virtual traffic between two virtual machines may never leave the physical host hardware, which

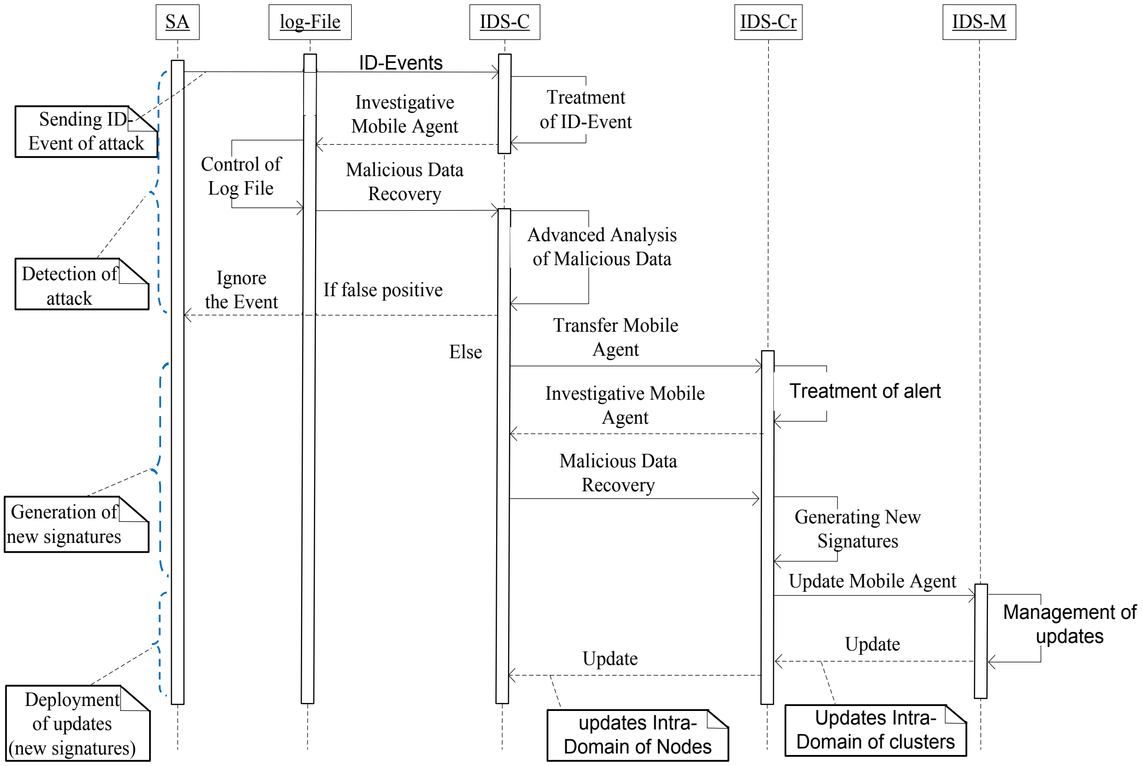


Fig. 3. Principle of our framework

use traditional physical firewalls unsuccessful to secure and monitor this traffic. Then, the best solution to this problem is the use of virtual firewalls over hypervisor. As shown in figure 4, a virtual firewall is a firewall service running in a virtualized environment, providing the usual packet filtering and monitoring services that a physical firewall would provide. Thus, we will have a hypervisor-based on virtual firewall. This virtual firewall is implemented on the VMM and it is responsible to capture malicious VM activities including packet injections. the implementation of the virtual firewall based on a modification to the physical host hypervisor kernel to add rules or modules allowing the VF system access to VM information and virtualized network interfaces moving packet traffic between VMs as well as and direct access to the virtual network switches. The Virtual Firewall can use the same features to then perform all firewall functions like forwarding, dropping and packet inspection, but without actually using the virtual network at any point.

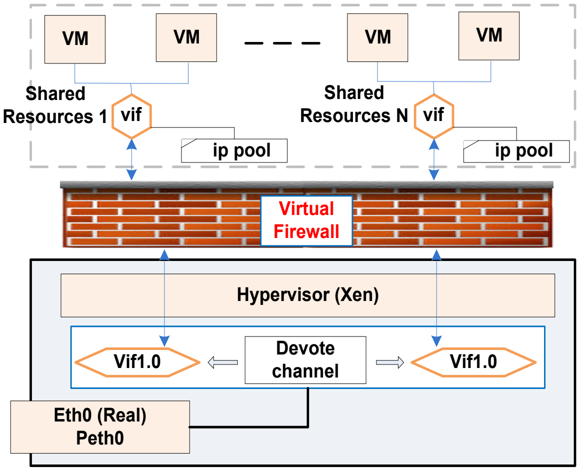


Fig. 4. Virtual firewall for securing the virtualized cloud computing infrastructure

IV. Discussion

Detection intrusion is major security concern in the Cloud. For ensure a high level of trust in cloud computing, we propose a new framework based on cooperative of Hy-IDS and mobile agents. This framework, allowed us to achieve three objectives, namely: intrusion detection at the front-end as well as the back-end of Cloud environment (i.e IaaS) of manner autonomous. Then, generating new signatures from malicious data or responses actions used by the firewall. Finally, dynamic deployment of updates between clusters in a cloud computing, using the newest signatures previously created. We used the signature generation algorithm and exchange of updates between clusters to achieve new knowledge and detect new kind of intrusion. About the virtual firewall, it receives its actions rules as signatures from IDS-M. Outstanding scalability is another strong point for this framework. When for example our VM migrates from server machine to another one (e.g. from Cluster-1 to Cluster-2), it is still possible to perform intrusion detection as our IMA can migrate just like VMs, and the same rule applies to other mobile agents (Transfer Mobile Agents and Mobile Agent Update). Moreover, this is the strength of our framework, which gives the IDS and NIDS great scalability and flexibility. Therefore, we have met almost all the mentioned challenges in our framework. Therefore, this framework has several advantages, it can be considered as an effective solution for the detection of intrusion into cloud computing. Thus, it can be used to protect people and property against risks of intrusion and aggression.

V. Conclusions

There is a major need of bringing security, transparency and reliability in cloud model for client satisfaction. Then, one of the security issues is how to reduce the impact of any type of intrusion in this environment. Thus in this paper, we propose an intelligent framework, which is based on the collaboration of the IDS-C, IDS-Cr, IDS-M and Mobile agents to detect attacks, and using the virtual firewalls to drop and block all types of attacks over hypervisor. As mentioned previously, mobile agents are used in our framework to investigate the VMs, transfer of malicious data, and exchange of update between different clusters in cloud computing.

Acknowledgment

We are thankful to God Almighty for giving us the knowledge and wisdom to complete this work. We are also thankful to our teachers and parents for their encouraging support.

References

[1] P. Mell and T. Grance, The NIST definition of cloud computing, National Institute of Standards and Technology, Gaithersburg, USA, 2011.

[2] Venkateshwaran K, Anu Malviya, Utkarsha Dikshit, S.Venkatesan. “Security Framework for Agent-Based Cloud Computing”, International Journal of Artificial Intelligence and Interactive Multimedia, Vol. 3, Nº 3. 2015

[3] Priyank Singh Hada Ranjita Singh Mukul Manmohan. “Security Agents: A Mobile Agent based Trust Model for Cloud Computing”. International Journal of Computer Applications, December 2011.

[4] A. Pandey, S. Srivastava. “An Approach for Virtual Machine Image Security”. International Conference on Signal Propagation and Computer Technology (ICSPCT), 2014.

[5] H. TOUMI, A. EDDAOUI and M. TALEA.” Cooperative Intrusion Detection System Framework Using Mobile Agents for Cloud Computing”. Journal of Theoretical and Applied Information Technology 10th December 2014. Vol.70 No.1

[6] H. TOUMI, A. TALEA, B. MARZAK, A. EDDAOUI, M. TALEA, “Cooperative Trust Framework for Cloud Computing Based on Mobile Agents”. International Journal of Communication Networks and Information Security (IJCNIS) Vol. 7, No. 2, August 2015.

[7] S. Roschke, Feng Cheng, C. Meinel. “Intrusion Detection in the Cloud”. Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009.

[8] Jean-Henry Morin, Jocelyn Aubert, Benjamin Gateau. “Towards Cloud Computing SLA Risk Management: Issues and Challenges”, 45th Hawaii International Conference on System Sciences, 2012.

[9] Y. Jadeja, K. Modi. “Cloud Computing - Concepts, Architecture and Challenges”. International Conference on Computing, Electronics and Electrical Technologies, 2012.

[10] Michael Armbrust , Armando Fox , Rean Griffith , Anthony D. Joseph , Randy Katz , Andy Konwinski , Gunho Lee , David Patterson , Ariel Rabkin , Ion Stoica , Matei Zaharia. “Above the Clouds: A Berkeley View of Cloud Computing”, UC Berkeley Reliable Adaptive Distributed Systems Laboratory, February 10, 2009.

[11] K. Benzidane, S. Khoudali, A. Sekkaki. “Autonomous Agent-based Inspection for inter-VM Traffic in a Cloud Environment”. The seventh International Conference for Internet Technology and Secured Transactions, 2012.

[12] A. Elsayed, N. Abdelbaki. “Performance Evaluation and Comparison of the Top Market Virtualization Hypervisors“. Eighth International Conference on Computer Engineering & Systems 2013.

[13] V.Nandgaonkar, A. B. Raut. “A Comprehensive Study on Cloud Computing”. International Journal of Computer Science and Mobile Computing, April- 2014.

[14] Yashpal Singh, Kapil Gulati, S. Niranjan.” Dimensions and issues of mobile agent technology”. International Journal of Artificial Intelligence & Applications (IJAIA), 2012.

[15] Chirag N. Modi, Dhiren R. Patel, Avi Patel, Muttukrishnan Rajarajan, «Integrating Signature Apriori based Network Intrusion Detection System (NIDS) in Cloud Computing». Second International Conference on Communication, Computing & Security. 2012

[16] Dastjerdi, Amir Vahid, Kamalrulnizam Abu Bakar & Sayed Gholam Hassan Tabatabaei. “Distributed Intrusion Detection in Clouds Using Mobile Agents”, In Proceedings of the 2009 Third International Conference on Advanced Engineering Computing and Applications in Sciences. ADVCOMP ’09 pp. 175–180, 2009

[17] Lin Chen, Xingshu Chen, Junfang Jiang, Xueyuan Yin, and Guolin Shao, “Research and Practice of Dynamic Network Security Architecture for IaaS Platforms”. Tsinghua Science and Technology. October 2014.

Hicham TOUMI received the MASTER degree in Network and Communication from Faculty of Sciences, CHOUAÏB DOUKKALI University El Jadida in 2013. He is preparing his PhD degree in the field of networks security in cloud computing using mobile agents approach at Faculty of Science Ben M’sik, MITI Laboratory, HASSAN II University.

Bouchra MARZAK was born in Casablanca, Morocco in 1989, received the MASTER degree in information processing from Faculty of Science Ben M’sik, Hassan II University Mohammedia-Casablanca in 2013. She is preparing her PhD degree in the field of clustering and dissemination data in vehicular networks at Faculty of Science Ben M’sik, MITI Laboratory, Hassan II University.

Amal TALEA is a graduate from Ecole Centrale Paris / specialized master in information system management. She also holds an engineering degree in computer and networking from National School of Engineer southern Alsace. Through her missions, she gained experience in project management, information systems governance, project portfolio management. Actually, she is preparing her PhD degree in Laboratory of Modeling and Information Technology, Department of Mathematics and Computer Science, Faculty of Sciences Ben M’sik, Hassan II University.

EDDAOUI Ahmed is a Professor-Researcher in Cloud Computing Security and resources management, Department of math and Computer sciences Faculty of sciences Ben M’Sik Hassan II University Morocco. PhD in Computer sciences (Information Security)

Mohamed TALEA was born in Casablanca, Morocco in 1964, Professor of Higher Education at the Faculty of Sciences Ben M’Sik, UNIVERSITY HASSAN II MOROCCO CASABLANCA. He obtained his PhD in collaboration with the LMP laboratory in Poitiers University, FRANCE in 2001. He obtained a Doctorate of High Graduate Studies degree at the University Hassan II in 1994. Actually, he is the Director of Information Treatment Laboratory. He has published twenty papers in conferences and national and international journals. His search major field is on Systems engineering, in security of system information.

Techniques to Detect DoS and DDoS Attacks and an Introduction of a Mobile Agent System to Enhance it in Cloud Computing

Abdelali Saidi[1], Elmehdi Bendriss[2], Ali Kartit[3], Mohamed El Marraki[1]

[1]LRIT associated unit to CNRST (URAC 29), Faculty of Sciences, Mohammed V University, PB 1014, Rabat, Morocco
[2]UFR SI3M, ENSIAS PB. 6624 - Al Irfane, Rabat 10112, Morocco
[3]LTI, departement TRI, ENSAJ Chouaib Doukkali University, El Jadida, Morocco

***Abstract* — Security in cloud computing is the ultimate question that every potential user studies before adopting it. Among the important points that the provider must ensure is that the Cloud will be available anytime the consumer tries to access it. Generally, the Cloud is accessible via the Internet, what makes it subject to a large variety of attacks. Today, the most striking cyber-attacks are the flooding DoS and its variant DDoS. This type of attacks aims to break down the availability of a service to its legitimate clients. In this paper, we underline the most used techniques to stand up against DoS flooading attacks in the Cloud.**

***Keywords* — Cloud computing; DoS DDoS; Mobile agent.**

I. Introduction

Cloud computing is, without any doubt, the future of IT systems. It brings along some advantages that can attract any type of companies. For example, Cloud gives high computing capabilities as a service (without buying the hardware) at a cheap cost, etc.

A. Cloud features

To be more attractive, the Cloud has to ensure the following features [1]:

- On-demand self-service: give the consumer the possibility to provision power of computing as needed without any human interaction;
- Broad network access: make the Cloud available from any type of network using any client platform;
- Resource pooling: the Cloud uses a multi-tenant model to serve multiple consumers. The resources have to be pooled to maximize the number of consumers;
- Rapid elasticity: make the consumers think that the resources are unlimited and available anytime they want more;
- Measured service: Cloud systems must monitor resources usage appropriate to the type of service. This can be done by using a metering capability.

B. Service models

To select a Cloud solution, the consumer must begin by deciding the appropriate service model. Following, the most popular services that Cloud offers:

- Software as a service (SaaS): the users can rent a set of applications running on the Cloud by the provider;
- Platform as a service (PaaS): the users have the service of implementing their applications on the Cloud and run it;
- Infrastructure as a service (IaaS): the users can rent a specific infrastructure from the Cloud and run any kind of applications even the operating system.

C. Deployment model

After the service model, the future consumer must think about how he would benefit from the Cloud. Here we have four models of the Cloud deployment:

- Private Cloud: The Cloud infrastructure will be used by a single consumer. The infrastructure can be maintained in the client's local or by a third party;
- Community Cloud: the Cloud will be used by a set of consonants clients that share a common interest. Also, the infrastructure can be deployed in the clients' locals like it can be managed by a third party;
- Public Cloud: the Cloud infrastructure is deployed by a Cloud provider for any client who wants to consume;
- Hybrid Cloud: is the composition of two or more deployment model.

II. Security issues in Cloud Computing

Basically, Cloud computing is a good IT infrastructure well maintained. Its main objective is to discharge clients from the infrastructure management. This will help the clients to focus only on their activities. However, besides security issues of IT systems, the Cloud brings some more specific issues.

A. Data security

In a traditional IT infrastructure, data is kept locally. And the owner does whatever it takes to ensure its confidentiality. Using the Cloud to store its data can seem doubtful since the client doesn't have any idea of how the data will be processed and where. Normally, the Cloud provider must ensure that even its own administrator won't have any way to reach the data or even log onto the clients' accounts.

B. Network security

When an organization trusts a Cloud provider, it must be aware of that the Internet will be used to transfer data from and to the Cloud. Internet is the most unpredictable network in the world; cyber-attacks are launched around the clock in it. Among the risks that threat every network communication we have:

- Packet Sniffing: it permits to intruders to analyze the traffic;
- Man in the Middle: it exploits a vulnerability in TCP/IP stack to deflect the traffic;
- IP Spoofing: it sends packets with a forged source IP address;
- Port scanning: it helps to detect network services running on a

DOI: 10.9781/ijimai.2017.4312

distant host;

- Network penetration: it permits to log on unauthorized session.

C. Data location

The first thing a potential Cloud client must do is to ask for a certification about the location where services will be stored. This can create a very annoying problem for the data confidentiality if the data is stored in a country where the regulation gives the right to some organizations to look onto the private data without the owner permission. For example, in the USA, USA PATRIOT ACT gives the government services access to data stored in any server.

Additionally, the Cloud is a multi-tenant system. It means that the computational resources will be used by many clients. The Cloud provider must ensure a perfect data isolation. Every client must be at ease regarding its data accessibility.

D. Web applications security

Accessing its Cloud requires a connection from the Internet and a terminal provided with a web client. It means that the applications deployed on the Cloud are mostly based on web platforms. This brings to the Cloud some issues related to the web shape. The open web application security project (OWASP) released a document about the ten most critical web application security risks [2]:

- Injection;
- Broken authentication and session management;
- Cross-site scripting;
- Insecure direct object references;
- Security misconfiguration;
- Sensitive data exposure;
- Missing function level access control;
- Cross-site request forgery;
- Using components with known vulnerabilities;
- Unvalidated redirects and forwards;

E. Virtualization issues

Since the virtualization is mostly used in the Cloud environments, it adds also some issues. The SANS institute have summarized some mistakes to avoid when using the virtualization [3]:

- Misconfiguring virtual hosting platforms, guests and networks;
- Failure to properly separate duties and deploy least privilege controls;
- Failure to integrate into change/lifecycle management;
- Failure to educate other groups, particularly risk management and compliance staff;
- Lack of availability or integration with existing tools and policies;
- Lack VM visibility across the enterprise;
- Failure to work with an open ecosystem;
- Failure to coordinate policy between VMs and network connections;
- Failure to consider hidden costs;
- Failure to consider user-installed VMs.

III. DoS and DDoS detection techniques

Several techniques have been proposed to mitigate DoS and DDoS attacks. These techniques can be classified into three types:

- Filtering techniques;
- Trace back techniques;
- Intrusion detection.

A. Hop-count filtering (HCF)

HCF is a filter dedicated to classify traffic based on the number of hops [4]. Initially, this filter has been used to handle IP spoofing attacks, but since most DoS attacks techniques send traffic with spoofed IP addresses, the filter can be useful also to detect DoS and DDoS attacks.

To calculate the number of hops that a packet has done before we receive it, we look into the TTL field. The value that we retrieve is simply the number of hops that the packet had the right to do before reaching its destination. To calculate the number of hops, we need to have an idea about its initial TTL value. According to [5] the initial value can be 30, 32, 60, 64, 128 or 255, and it depends on the operating system. And since the diameter on the Internet between two distant terminals rarely exceeds 30 hops [6] we can say that the initial TTL value is the smallest initial value that is superior to the received TTL.

Mukaddam et al [7] have enhanced this technique by adding a new parameter: RTT (Round Trip Time) to the considerations of the filter. This parameter give to the filter the possibility to make the difference between some packets that have done the same number of hops but different times of the round trip. Another enhancement of this method was given by Wang et al [8] proposing to implement the filter on the gateways. Maheshwari et al [9][10] underline the problem of computing time because of the large amount of packets that the filter can receive. To adjust this problem, they propose a technique called DPHCF-RTT that will create collaboration between the gateways and taking the different initial TTL values that may have been used into consideration.

B. Confidence based filtering (CBF)

CBF is a technique that helps to detect every deviation of the traffic from its normal shape [11]. It is also an enhancement of the HCF technique which considers different fields in the packet. CBF is based on some correlation between these fields that can be noticed after a period. This correlation builds a normal profile and the CBF filter tries to detect every deviation from it.

Fig. 1 illustrates how the CBF filter works.

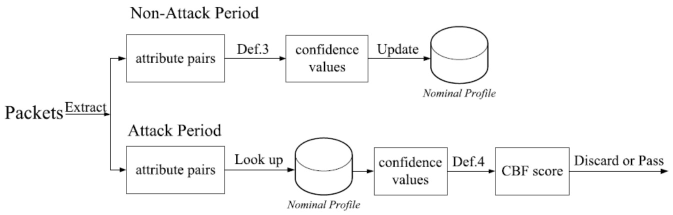


Fig. 1. CBF filter working

Priyanka et al [12] propose an enhancement of this filter by adding a new field in the packet header. This field will be called “confidence value” and it will be filled by every gateway it passes by. This will eventually give a modification into the IHL field. Mamtesh et al [13] propose an enhancement where the CBF will work neighboring a HCF filter.

C. Random port hopping (RPH)

RPH is a technique that permits to a server to change the port number when communicating with a legitimate client. Firstly, this technique was used to sidetrack spies. Lee et al [14] used this technique to mitigate DoS and DDoS attacks.

To guess the port number on which the server will wait for a packet, the client and the server must pre-share a key and divide time to slots. In the beginning of every slot, the client has to calculate the port number using an algorithm and the pre-shared key.

Zhang et al [15] propose a solution to make the server able to communicate with different clients in the same time. Lu et al [16] supported this technique by studying the rate of success when detecting attacks.

D. IP traceback

IP traceback is a technique that permits to track down spoofed packets to determine their true origin. There are different ways to track back a packet:

- Link testing: this method begins from the default gateway of the attack target and tries to detect the previous hop one by one recursively. This method consider that the attack remains active until we find the originator;
- Logging: this method tries to log every packet that has passed through a key gateway from the Internet. With this technique, the attack may be detected even if the attacker had finished. But, we have to consider the resources that will be consumed on the routers just to log the packets;
- ICMP traceback: Every router will randomly take a packet from 20 000 one and send an ICMP message to the owner of the destination IP address. This will help the destination to have an idea about the route that a packet has taken before being received.

E. Mobile agent techniques

A mobile agent is a program which can move from a computer to another autonomously and continue its execution. It brings some advantages that make mobile agents suitable for building intrusion detection systems like:

- Computation bundles:
- Parallel processing:
- Dynamic adaptation:
- Tolerant to network faults:
- Flexible maintenance:

Several works adopt mobile agents to face DoS and DDoS attacks. Akyazi and Uyar proposed four methods; three of them use mobile agents [17]. Each method use Snort like a sensor. The contribution of the mobile agent platform is reducing bandwidth usage by moving data analysis near to the source of the intrusion data. Zamani and al [18] propose a mobile agent platform inspired from danger theory to build an intrusion detection system resilient to DDoS attacks. This system represents a model of immunization of distributed intrusion detection system. Armoogum and mohamudally [19] underlined the issues of most IP traceback solutions such as high false positives, enormous storage requirements at routers and huge additional data in network traffic. To mitigate these problems, a mobile agent platform was proposed for real-time traceback of distributed attacks. Demir and al [20] proposed an enhancement of this type of solutions by demonstrating how a careful placement of agents can improve an earlier DDoS detection.

IV. A mobile agent system to enhance DoS and DDoS detection in Cloud Computing

Our mobile agent system begins by classifying virtual machines into several sets. Each set of VMs will be monitored by a mobile agent. Eventually, the number of VMs in a set affects the quality of the mobile agent service. The mobile agent has to move from a virtual machine to another following a priority metering capability. This capability helps to define the order of VMs but can be broken if one of them is in a critical situation. Thus, we placed sensors in every VM to keep eye on the hardware usage. If a sensor detects an overtaking it sends an event to the mobile agent. This latter will have to move to this VM and the order will be reset [21].

A. VMs order

When the mobile agent of a set of VMs moves, it will choses the next VM following the order of priority. Assuming a set having five VMs (A, B, C, D and E) with priorities respecting that order. If the mobile agent is currently working on B and receives an event from D the mobile agent will move to B and the order will be like:

D -> B -> C -> E -> A

Thus, the mobile agent choses the next VM based on the time of last visit.

B. Components of the mobile agent

The mobile agent must analysis a VM, decide if something wrong in it, respond a proper reaction and be aware of some states of the other VMs. To handle all of this, the mobile agent must contain the following components:

- Listening module: this part of the mobile agent will receive the sensors traps;
- Analyze module: this one studies information in the environment logs to find suspicious data;
- Decision module: this one compares the suspicious data the attacks' scenarios we have in our database to decide if it is an attack;
- Response module: this latter chose the best reaction to execute automatically in order to limit damages.

C. Scenarios

Different scenarios that our mobile agent can handle:

- Multiple VMs fall in critical condition simultaneously: when the mobile agent is proceeding in VM which is in a critical condition and receives another trap from another one, the mobile agent will create another instance of it and send it to this latter VM. Every agent clone must send the result of its work to the parent agent.
- A distributed attack that targets multiple VMs in deferent sets: if there is a malicious data that can be part of coordinated attack reaching VMs in different sets, the agent who suspects this must send a trap to every mobile agent. These latters have to take this in consideration.

V. Conclusion

In this paper, we presented valuable works on the detection of DoS and DDoS. We noted the "HCF filter" and its generalization "the CBF filter" that both try eliminate packets with spoofed IP addresses, the RPH that try to divert the attacker, the IP traceback that tries to detect the source of the attack and the mobile agent systems that try to give another way to detect DoS and DDoS attacks in the Cloud. Then, we introduced our mobile agent system and depicted its way to handle things for different VMs. We are still working on the implementation of our solution and studying other scenarios that the mobile agent can run into.

References

[1] National Institute of Standards and Technology. « The nist definition of cloud computing ». 2011.

[2] OWASP. « The ten most critical web application security risks ». 2013. http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf

[3] SANS Institute. « Top Virtualization Security Mistakes (and How to Avoid

Them) ». 2009. https://www.sans.org/reading-room/whitepapers/analyst/top-virtualization-security-mistakes-and-avoid-them-34800

[4] Cheng Jin, Haining Wang, and Kang G. Shin. Hop-count filtering: "An effective defense against spoofed ddos traffic". In Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS '03, pages 30–41, New York, NY, USA, 2003. ACM

[5] The Swiss Education and Research Network. "Default ttl values in tcp/ip". 2002.

[6] Bill Cheswick, Hal Burch, and Steve Branigan. Mapping and visualizing the internet. In Proceedings of the 2000 USENIX Annual Technical Conference, pages 1–12, 2000.

[7] A. Mukaddam and I.H. Elhajj. Round trip time to improve hop count filtering. In Broadband Networks and Fast Internet (RELABIRA), 2012 Symposium on, pages 66–72, May 2012.

[8] Xia Wang, Ming Li, and Muhai Li. A scheme of distributed hop-count filtering of traffic. In Wireless Mobile and Computing (CCWMC 2009), IET International Communication Conference on, pages 516–521, Dec 2009.

[9] R. Maheshwari, C.R. Krishna, and M.S. Brahma. « Defending network system against ip spoofing based distributed dos attacks using dphcf-rtt packet filtering technique ». In Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on, pages 206–209, Feb 2014.

[10] C.R. Krishna R. Maheshwari. « Mitigation of ddos attacks using probability based distributed hop count filtering and round trip time ». International Journal of Engineering Research and Technology (IJERT), 2(7), 2013.

[11] Wanchun Dou, Qi Chen, and Jinjun Chen. A confidence-based filtering method for ddos attack defense in cloud environment. Future Gener. Comput. Syst., 29(7):1838–1850, September 2013.

[12] Priyanka Negi, Anupama Mishra, and B. B. Gupta. « Enhanced CBF packet filtering method to detect ddos attack in cloud computing environment ». CoRR, abs/1304.7073, 2013.

[13] Mamtesh and Rajender Nath. "An improved defense mechanism based on packet filtering to mitigate ddos attack in cloud computing environment". IJCA, 5, 2015.

[14] H.C.J. Lee and V.L.L. Thing. « Port hopping for resilient networks ». In Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th, volume 5, pages 3291–3295 Vol. 5, Sept 2004.

[15] Zhang Fu, M. Papatriantafilou, and P. Tsigas. « Mitigating distributed denial of service attacks in multiparty applications in the presence of clock drifts ». Dependable and Secure Computing, IEEE Transactions on, 9(3):401–413, May 2012.

[16] Yue-Bin Luo, Bao-Sheng Wang, and Gui-Lin Cai. Effectiveness of port hopping as a moving target defense. In Security Technology (SecTech), 2014 7th International Conference on, pages 7–10, Dec 2014.

[17] U. Akyazi and A.S.E. Uyar. « Distributed intrusion detection using mobile agents against ddos attacks ». In Computer and Information Sciences, 2008. ISCIS '08. 23rd International Symposium on, pages 1–6, Oct 2008.

[18] M. Zamani, M. Movahedi, M. Ebadzadeh, and H. Pedram. « A ddos-aware ids model based on danger theory and mobile agents ». In Computational Intelligence and Security, 2009. CIS '09.International Conference on, volume 1, pages 516–520, Dec 2009.

[19] M. Duraipandian and C. Palanisamy. « An intelligent agent based defense architecture for ddos attacks ». In Electronics and Communication Systems (ICECS), 2014 International Conference on, pages 1–7, Feb 2014.

[20] O. Demir, B. Khan, G. Ben Brahim, and A. Al-Fuqaha. « Optimizing agent placement for flow reconstruction of ddos attacks ». In Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, pages 83–89, 1July 2013.

[21] Abdelali Saidi, El Mehdi Bendriss, Ali Kartit and Mohamed El Marraki. "A Mobile Agent System to Enhance DoS and DDoS Detection in Cloud Computing". European Journal of Scientific Research. Volume 131 No 2. April, 2015.

Abdelali Saidi is a PhD Student in Networks Security at Mohammed V University in Morocco since 2011. He has a Master degree in Systems and Computer Networks from Ibn Tofail University. He teaches in the area of computer science (Linux, IP Networks, and Information Security) since 2012. His main interest is Cloud Computing Security researches. abdelali.saidi@gmail.com

Elmehdi Bendriss received his PhD degree in Networks Security from ENSIAS, Mohamed V University in 2014. He also holds MSC in IT from the same University and an engineering diploma from INPT since 2002. He's now working on systems and networks Security and especially in Cloud computing. He's been teaching in the area of Computer Science (Systems and Networks administration, Information Security, Virtualization) since 2003. bendriss@gmail.com

Ali Kartit received the PhD degree in Computer Science (November 2011) Specialty Security of Computer Networks. He graduated from the University Mohamed V Faculty of Rabat. Now, He works as an assistant professor at the University Chouaib Doukkali of El Jadida. The author has developed a rich and diverse experience of over 14 years in the computer world, including 10 years in technical and vocational education as a computer network trainer and manager module ”computer network security notions” and 4 years in the corporate world as Administrator of computer networks and Head of the park. The author is a certified Cisco and Microsoft Exchange Server 2003. His research area covers security policies of firewalls, the Intrusion detection systems (IDS) and cloud security. alikartit@gmail.com

Mohamed El Marraki received the Doctorate and the Doctorate of the State degrees in algebra and number theory, respectively, from the Bordeaux University, France in 1991, and the Mohammed V-Agdal University, Rabat, Morocco, in 1996; he also received the Doctorate in "dessin d'enfant theory" from the Bordeaux University, France in 2001. He joined Mohammed V University, Rabat, Morocco, in 1996, first as an associate professor and full Professor since 2000, where he is teaching. Over 19 years, he developed teaching and research activities covering various topics of Mathematics, cryptography and graph theory which allow him to advise 5 PhD theses and publish over 60 journal papers and conference communications. Mohamed El Marraki is member of the several "Scientific Program Committee" of the International conference. He is a member of several mathematical and computer science journals. marraki@fsr.ac.ma

IJIMAI

http://ijimai.unir.net