Volume 3, Issue 2
November 2016
ISSN: 1390-9266

LAJC

LATIN-AMERICAN JOURNAL OF COMPUTING

FACULTAD DE INGENIERÍA DE SISTEMAS
QUITO - ECUADOR

http://lajc.epn.edu.ec/

ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS

LATIN AMERICAN JOURNAL OF COMPUTING
LAJC

Vol III, Issue 2, November 2016
ISSN: 1390-9266
e-ISSN: 1390-9134

Published by:
Escuela Politécnica Nacional
Facultad de Ingeniería de Sistemas

Quito – Ecuador

LATIN AMERICAN JOURNAL OF COMPUTING – LAJC

Published by:

Escuela Politécnica Nacional
Facultad de Ingeniería de Sistemas
Ecuador

Mailing Address:

Escuela Politécnica Nacional, Facultad de Ingeniería de Sistemas
Ladrón de Guevara E11-253, La Floresta
Quito-Ecuador, Apartado Postal: 17-01-2759

Web Address:

http://lajc.epn.edu.ec

E-mail:

lajc@epn.edu.ec

Frecuency:

2 issues per year

Circulation:

350

EDITORIAL

A tree is strong because of its roots

Every single day, it is possible to identify those anonymous actors who, far from receiving recognition, make our society work. Their role is, usually, unnoticed with exception of those situations when something goes wrong. We can see clear examples of that in our lives. For instance, consider the staff responsible for ensuring the supply of water to our homes. We never have a glimpse of their sacrificed work unless water stops running of our shower. A situation that, by the way, usually happens when we are soaping.

The same thing happens with academic journals. In general, we do not think about all the activities involved in publishing the articles that we read. It is true that the weight of scientific production lies with the authors. However, there are other actors, the reviewers, who usually receive no credit, but without their work, the quality of a paper could not be assured. Hidden in the shadows, the role of reviewers for the maturation of an article is almost always unknown.
Peer review is a critical part of the functioning of a scientific community. Its role is relevant for quality control of contributions. Thus, a journal committed on the broadcasting of articles with high quality scientific content, such as the Latin American Journal of Computing, owes its achievements to a great extent to the work done by its anonymous, invisible but invaluable reviewers.

Review a paper is not simple. The reviewer must go in depth of the article. He must understand the view given by the authors, the problem raised, the methodology followed, the discussion of the results, and the subsequent conclusions. He should consult the references and update his knowledge on the subject by reading other complementary sources. He must also identify the elements to be corrected and, above all, to offer suggestions to authors for maturing the paper a little more. This means that their work goes beyond mere arbitration, but it includes a final pre-publication guidance. A workload that is not insignificant and by which the reviewers receive, as rewards, self-satisfaction and the gratitude of the editorial board.
Although, when a reviewer accepts the assignment of review an article, he accepts to work under conditions of anonymity and volunteering. This is the reason why the editorial board wants to recognize our collaborators through the creation of the “Annual LAJC Best Reviewer Award.” This prize that will be granted based on three criteria:

- The quality of the analysis of the article, globally and by sections, that allow identifying the coherence, strengths and weaknesses of the paper.
- The quality of suggestions, in terms of both the form and the content, that allow the evolution of the article towards a clear and significant contribution of knowledge.
- The respect of delays and the overall evaluation process.

With the prize, the editorial board does not pursue to incentive competition nor to impose standardized reviewing models. The objective is to shed light on the efforts of those whose work is indispensable for our magazine.

The articles we publish in this issue are all good examples of the added value from our contributors. They all passed through a blind review of two reviewers. As a result of the review, the authors received suggestions that allowed them to improve their paper and being accepted for publication.

The article written by Juan Vizcarrondo, José Aguilar, Ernesto Exposito and Audine Subias presents a component of knowledge management build over the base of different paradigms.

The component complements a distributed architecture for fault self-correction in the composition of services. This paper continues previous efforts of the same authors.

The article by Maribel Mondoça, José Aguilar and Niriaska Perozo presents a reflective middleware to facilitate the emergence of ontologies in intelligent environments. The proposed architecture is based on the autonomic computing model, which allows to satisfy the consistency and semantic evolution that each environment would require.
The paper by Miguel Flores, Guido Saltos and Sergio Castillo proposes a model for the classification of DNA sequences for different types of cancer. The model focuses on classifying the expression levels of certain genes that are present in cancerous tumors. The article includes the results of an exploratory study where an analysis of depth measurements for several types of cancer was carried out.

Julián Galindo's article presents an alternative method for software programming that combines storyboards with the spiral development model. The applicability of the method is illustrated with the case of a virtual store. The resulting application was evaluated using the concepts of the technology acceptance model. This method can be used mainly by small software companies interested into combine HCI with formal methods of software development, while they guarantee the acceptance of the end users.

Special mention should be made of the selected articles of the VII JISIC 2016 whose authors are: Francisco Bolaños, Luis García and Antonio Cevallos; and Jesennia Iñiguez, Rene Guaman, Robert Figueroa and Freddy Ajila. This year, by decision of the editorial board, two articles of JISIC 2016 were accepted for publication and there was not a special issue of the conference. Other articles, however, were shortlisted for publication in the next issue. The authors received recommendations from a reviewer of the journal to improve their work. These recommendations complement those received by the reviewers of the conference.

Finally, we would like to reiterate our gratitude to the reviewers of our journal. They work in universities of Argentina, Australia, Brazil, Colombia, Ecuador, Spain, France, Japan and Venezuela. It is thanks to their work that our magazine grows every day.

We hope that the articles in this issue meet the expectations of our readers. Good reading.

A diario, es posible identificar esos personajes anónimos que, lejos de recibir un merecido reconocimiento, son quienes permiten realmente que nuestro mundo funcione. Su rol suele pasar desapercibido, a excepción de aquellas ocasiones en las cuales percibimos que algo no funciona bien. Ejemplos hay varios en nuestra sociedad. El personal encargado de garantizar el suministro de agua a nuestros hogares es uno de ellos. Nunca tenemos un atisbo de su sacrificado trabajo a menos que el agua deje de salir de nuestra ducha. Situación que, por cierto, suele coincidir con el momento en el que estamos enjabonados.

Lo mismo sucede con las revistas científicas. En general, no pensamos en todas las actividades involucradas en llevar los artículos incluidos en el número que llega a nuestras manos. Es verdad que el peso de la producción científica recae sobre los autores. Pero existen otros personajes, los revisores, que no suelen recibir crédito alguno, pero sin cuya labor la calidad científica de una publicación no podría ser asegurada. Escondidos en la sombra, el rol de los revisores en la maduración de un artículo pasa casi siempre desconocido.
La revisión por pares es la parte fundamental del funcionamiento de una comunidad científica. Su importancia es relevante en el control de calidad de una publicación. Y una revista

comprometida con la difusión de artículos con contenido científico relevante, como el Latin American Journal of Computing, debe sus logros en gran medida a la labor ejercida por sus revisores anónimos, invisibles pero invaluables.

El trabajo de revisión no es simple. Un revisor debe inmiscuirse en profundidad en el artículo que revisa. Debe comprender el enfoque dado por los autores, la problemática planteada, la metodología adoptada, los elementos de discusión expuestos y las conclusiones resultantes. Debe revisar las referencias citadas y actualizar sus conocimientos sobre la materia acudiendo a otras fuentes complementarias. Debe además identificar los elementos a corregir y, sobretodo, transmitir sugerencias a los autores para que el artículo madure un poco más. Esto significa que su labor va más allá del simple arbitraje, incluye una forma de guía final previa a la publicación. Una carga de trabajo que no es insignificante y por la cual los revisores reciben, como única recompensa, la autosatisfacción por el aporte dado a los autores y el agradecimiento del comité editorial.

Si bien, al momento de aceptar el encargo de revisar un artículo, un académico acepta trabajar las condiciones del anonimato y voluntariado. Como miembros del comité editorial, no podemos dejar pasar la oportunidad de realizar un reconocimiento a nuestros colaboradores a través de la creación del "Premio anual al mejor revisor de la revista LAJC". El mismo que será concedido en base a tres criterios:

- La calidad del análisis del artículo, a la vez global y de cada una de sus partes, que permita identificar la coherencia, fortalezas y debilidades del mismo.
- La calidad de las sugerencias de mejora, tanto en la forma como en el fondo, que permitan la evolución del artículo hacia una contribución clara y significativa de conocimientos.
- El respeto de los periodos y del proceso de evaluación.

Lejos de que este premio incite a la competencia y menos aún que el mismo pretenda imponer modelos de evaluación, el premio busca simplemente arrojar luz sobre aquellos que trabajan lejos de las luces y cuya labor es indispensable para nuestra revista.

Los artículos que publicamos en este número son todos buenos ejemplos del valor agregado que nuestros colaboradores aportan. Todos ellos pasaron por una revisión ciega de al menos dos revisores. Y como resultado, los autores recibieron recomendaciones que les permitieron mejorar sus trabajos y obtener la aceptación del comité editorial para publicación.

El artículo de Juan Vizcarrondo, José Aguilar, Ernesto Exposito y Audine Subias presenta un componente de gestión de conocimientos sobre la base de diferentes paradigmas. El componente presentado complementa una arquitectura distribuida para auto-corrección de fallas en la composición de servicios. Este trabajo se inscribe en la continuidad de contribuciones anteriores de los mismos autores.

El artículo de Maribel Mondoça, José Aguilar y Niriaska Perozo presenta un middleware reflexivo para facilitar el surgimiento de ontologías en ambientes inteligentes. Las ontologías generadas incorporan conceptos provenientes del análisis de necesidades del ambiente inteligente y de información de contexto. La arquitectura propuesta se basa en el modelo de computación autonómica que permite mantener la consistencia y evolución semántica que el ambiente requiera.

El artículo de Miguel Flores, Guido Saltos y Sergio Castillo propone un modelo para la clasificación de secuencias DNA para diferentes tipos de cáncer. El modelo se focaliza en

clasificar los niveles de expresión de ciertos genes presentes en tumores cancerígenos. El artículo incluye los resultados de un estudio exploratorio donde se realiza un análisis de medidas de profundidad para varios tipos de cáncer.

El artículo de Julián Galindo presenta un método alternativo de desarrollo de aplicaciones que combina guiones gráficos con el modelo de desarrollo de software en espiral. La aplicabilidad del método es ilustrada con el caso del desarrollo de una tienda virtual. La aplicación resultante fue evaluada mediante el uso de los conceptos del modelo de aceptación de tecnología. El método presentado puede ser utilizado principalmente por pequeñas compañías de desarrollo de software que busquen combinar técnicas HCI con métodos formales de desarrollo mientras garantizan la aceptación de los utilizadores finales.

Mención especial merece los artículos seleccionados de las VII Jornadas de Ingeniería de Sistemas Informáticos y Computación JISIC 2016 cuyos autores son: Francisco Bolaños, Luis García y Antonio Cevallos; y Jesennia Iñiguez, Rene Guamán, Robert Figueroa y Freddy Ajila. Este año, por decisión del consejo editorial, se aceptaron dos artículos de JISIC 2016 para publicación y no hubo número especial de las jornadas. Otros artículos, sin embargo, fueron preseleccionados para publicación en el próximo número. Los autores recibieron recomendaciones de un revisor de la revista para mejorar sus trabajos. Estas recomendaciones complementan las que recibieron por parte de los revisores de las jornadas.

Finalmente, deseamos reiterar nuestro agradecimiento al cuerpo de revisores de nuestra revista, quienes trabajan en instituciones de Argentina, Australia, Brasil, Colombia, Ecuador, España, Francia, Japón y Venezuela. Es gracias a su trabajo que nuestra revista crece cada día.

Esperamos que los artículos de este número suplan las expectativas de nuestros lectores. Buena lectura.

Edison LOZA AGUIRRE
Université de Grenoble Alpes
Editorial Board LAJC

Latin American Journal of Computing – LAJC

Reviewers

We are most grateful to the following individuals for their time and commitment to review manuscripts for Latin American Journal of Computing – LAJC.

Aguiar Pontes Josafá, PhD. Tokyo Institute of Technology, Japan
Aguilar José, PhD. Universidad de los Andes, Venezuela
Anchundia Carlos, MSc. Escuela Politécnica Nacional, Ecuador
Andrade Roberto, MSc. Escuela Politécnica Nacional, Ecuador
Barriga Jhonattan, MSc. Escuela Politécnica Nacional, Ecuador
Benalcázar Marco, PhD. Escuela Politécnica Nacional, Ecuador
Brandão Diego, PhD. Universidade Federal Fluminense, Brasil
Buitrago Alex, PhD. Universidad Externado de Colombia, Colombia
Calle Tania, MSc. Escuela Politécnica Nacional, Ecuador
Carrera Iván, MSc. Escuela Politécnica Nacional, Ecuador
Carrión Gordón Lucía, MSc. University of Technology Sidney, Australia
Duarte Ferreira Vera Lúcia, PhD. Universidade Federal do Pampa, Brasil
Flores Pamela, PhD. Escuela Politécnica Nacional, Ecuador
Flores Denys, MSc. University of Warwick, England
Fuertes Díaz Walter, PhD. Universidad de las Fuerzas Armadas - ESPE, Ecuador
García Olaya Angel, PhD. Universidad Carlos III de Madrid, España
Hallo María, MSc. Escuela Politécnica Nacional, Ecuador
Hernández Myriam, PhD. Escuela Politécnica Nacional, Ecuador
Herrera Juan, MSc. Escuela Politécnica Nacional, Ecuador
Intriago Monserrate, MSc. Escuela Politécnica Nacional, Ecuador
Loza Aguirre Edison, PhD. Université de Grenoble, France
Lucio José Francisco, PhD. Escuela Politécnica Nacional, Ecuador
Luján Mora Sergio, PhD. Universidad de Alicante, España
Magreñán Ángel Alberto, PhD. Universidad Internacional La Rioja, España
Meliá Beigbeder Santiago, PhD. Universidad de Alicante, España
Navarrete Rosa, MSc. Escuela Politécnica Nacional, Ecuador
Paz Arias Henry, MSc. Escuela Politécnica Nacional, Ecuador
Pérez María, PhD. Escuela Politécnica Nacional, Ecuador
Pousa Federico, PhD. Universidad de Buenos Aires, Argentina
Qazi Farrukh, University of Warwick, England
Ramió Jorge, PhD. Universidad Politécnica de Madrid, España
Roa Marin Henry, MSc. The University of Queensland, Australia
Sánchez Gordón Sandra, MSc. Escuela Politécnica Nacional, Ecuador
Suntaxi Gabriela, MSc. Escuela Politécnica Nacional, Ecuador
Torres Olmedo Jenny, PhD. Escuela Politécnica Nacional, Ecuador
Yacchirema Diana, MSc. Universitat Politécnica de València, España
Zambrano Patricio, MSc. Escuela Politécnica Nacional, Ecuador

TABLE OF CONTENTS

The Component of Knowledge Representation of ARMISCOM for the Self-healing in Web Services Composition

J. Vizcarrondo, J. Aguilar, E. Exposito, A. Subias

***Abstract*— A previous work has proposed a reflective middleware Architecture for the management of service-oriented applications. Our middleware is designed to be fully distributed through all services of the SOA Application. The architecture uses the model of Autonomic Computing which allow the adaptation of our system, in order to self-healing. Particularly, one of the main aspects of this architecture is the representation of the knowledge. Our architecture uses different paradigms for the representation of the knowledge. For the diagnosis task, it uses chronicles, and for the reparation task it uses ontologies. In this paper, we present the knowledge representation framework, which represents the knowledge needed to perform the different operations of the middleware. Specifically, we design a distributed knowledge based on distributed chronicles, ontologies and other data structures.**

***Index Terms*—Web service fault tolerance, service composition, fault-repair ontology, Distributed Pattern Recognition, Reflective middleware**

I. INTRODUCTION

The SOA applications (Service Oriented Architecture) are flexible distributed applications, with loose coupling between these components, based on a software development model composed of small units, called services, which operate in heterogeneous distributed environments. This approach encourages a programming style based on the composition and reuse of services (new applications based on existing services).

The Services are inherently dynamics [1] because they can evolve (their internal calculation, interfaces, among others) and alter its results. Now, in the service composition, a failure of a single service generates an error propagation in the other services, and in this way, the failure of the system. Such failures are very hard to be detected and located, so it is necessary to develop new approaches to enable the diagnosis and correction of the fails, locals (in a service) or global (in the composition)

One of the main aspects to solve in SOA applications is their fault tolerance. For that is required a reparation procedure (self-healing). Repair is to restore the broken functionality, and to return the system at the normal execution [20, 21]. Correction of faults in web services always depends of the type of fault. Web service faults can be classified at three levels [2]: physical, development and interactions; additionally, each fault type has a different repair mechanism.

A previous work has proposed a distributed architecture for the self-healing of faults in the services composition, called ARMISCOM (Autonomic Reflective MIddleware for management of Services COMposition). In ARMISCOM, the fault diagnosis is carried out between the diagnosers present in each service [17]. Similarly, repair strategies are developed through consensus among distributed repair services. In this paper, we present *the* knowledge *representation* component of ARMISCOM, which represents the knowledge needed to perform the different operations of the middleware; specifically, it is the knowledge required by the analyzer and planner components of ARMISCOM.

II. RELATE WORKS

There are two types of failures in web services, the faults in a service, and the faults in the sequence of calls in a composition of services. In [2] is proposed a taxonomy of failures in web services, and describes the perceived effects. In addition, they propose a correlation of the failures and the reparation mechanisms. In [9, 10] propose other classification of Fault types, and define the Recovery action of each one.

At the level of architectures for fault management and recovery of the web services composition, [3] proposes a reflective middleware, called SOAR, which is designed as a centralized structure, in order to monitor and adapt the web application. The middleware has two levels: the first describes the basic characteristics of a SOA system (base level), and the second monitors and adapts the SOA system (meta level). The reflective part of the middleware executes the dynamic binding of web services composition, connecting or disconnecting the services of the SOA application.

In [5] is defined a decentralized architecture that has 2 levels. The first level defines a local diagnoser for each service of the composition. The second level is composed of a global diagnoser, which coordinates the local diagnosers to analyze the

Dr Aguilar has been partially supported by the Prometeo Project of the Ministry of Higher Education, Science, Technology and Innovation of the Republic of Ecuador.

J. Vizcarrondo is with Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL), Mérida – Venezuela. (email: jvizcarrondo@cenditel.gob.ve)

J.L. Aguilar is with CEMISID, Universidad de Los Andes, Mérida, Venezuela. Additionally, it is Prometeo Researcher at the Escuela Politécnica Nacional, Quito, and the Universidad Técnica Particular de Loja, Ecuador (aguilar@ula.ve)

E. Exposito and A. Subias are with CNRS, LAAS, 7, avenue du Colonel Roche, F-31400, Univ de Toulouse, INSA, F-31400, Toulouse - France (email: {ernesto.exposito,subias}@laas.fr})

failures. The global diagnoser also implements the mechanisms for the composition recovery. Each local diagnoser has chronicles that describe the failure patterns, and communicates their instantiations to the global diagnoser. The global diagnoser calculates the sequence of events in the service, to find the occurrence of an error, according to the chronicles instanced by the local diagnosers.

[4] proposes a centralized architecture for web services reparation. The architecture is composed of three modules: a module for monitoring and measuring (it determines the QoS parameters that are relevant), a module of diagnosis and definition of strategies (it detects the degradation of the system and builds the reparation plans), and a module of reconfiguration (it executes the reparation plan). Also, in [6] is proposed other centralized architecture, based on QoS monitoring. Furthermore, in [22] is proposed a structure composed of local diagnosers, which are coordinated by a global diagnoser that executes the repair tasks.

In the context of autonomic computing, MAPE has been used to manage failures in web services [11] providing the ability to self-healing in its invocation (alone web services), but not considering failures derived in its composition with other services. Also, other architecture based on MAPE has been proposed to study the faults on the services composition [12], but this architecture is completely centralized.

Recently, in [17] we proposed a reflective middleware architecture for fault management in service composition, called ARMISCOM, in which each service is overseen by a Local diagnoser using chronicles. To complete the proposal, this paper proposes the knowledge representation component of ARMISCOM. The knowledge representation component is responsible for the management of the knowledge base required by our middleware to carry out its Self-healing task.

The Knowledge representation component of ARMISCOM is composed by distributed chronicles, an ontology to correlate faults and repair methods, and a metadata for storing repair methods available for services within the composition. In previous works, we have designed the distributed chronicles and implemented a mechanism for the recognition of the distributed chronicles using the IEP component in OpenESB and the CQL language [19]. In this paper, we present in detail the design of a distributed ontology in order to correlate the fault type in services with the repair methods, based on [2], which can be used to make inferences about the functional and non-functional properties of the flows in the composition. Additionally, because various repair methods can be applied to solve a given failure, not all can be applied in a given moment because they are not available, is why, in this paper, we also define a distributed data structure for storing the possible repair methods that can be applied at any given time. In this way, this paper presents the design of the component of the distributed Knowledge representation of ARMISCOM for its operation, in order to be used in the self-healing of the web service composition, which contrast with the commonly used mechanisms based on centralized architectures.

III. ARMISCOM ARCHITECTURE

ARMISCOM is a reflective middleware architecture for faults management in the services composition [17]. Reflection is the ability of our middleware to monitor and modify their own behavior, as well aspects of its implementation (syntax, semantics, etc.), allowing the ability to be sensitive to their environment. Thus, ARMISCOM has a dynamic and adaptive behavior, fully distributed, in order to have a closer view of the occurrence of the events that occur in the application. ARMISCOM is divided into two levels, like classic reflective middlewares (see Fig. 1) [17]:

- **Base Level:** A services composition is defined as a set of calculations and interactions of the services that compose a SOA application, with a set of rules that determines these interactions. The base level knows the interactions and its rules in the choreography. In addition, the base level observes both the SOA system and the SOA application. In specific, it monitors the WSDL, UDDI, OWL-S and SCA elements of a SOA system, and uses FraSCAti platform for the intersection process of the services choreography.
- **Meta Level:** it provides the capacity of reflection. It analyses the message exchange between the services that are part of the composition and the components of the SOA system, in order to carry out the introspection. There is a meta level in each service of the choreography.

The implementation of ARMISCON has been designed based on the autonomic computing paradigm. The Autonomic Computing is a computing model inspired on the self-management in the autonomic nervous system of the human [7]. This system is capable of self-administer, for which defines an architecture consisting of 6 levels [7]:

- **Managed Resource:** is any resource of hardware or software.
- **Touch Point:** has the sensor and/or actuator mechanisms.
- **Autonomic Manager:** has the intelligent control loop, with the tasks automate the self-regulation of the applications. The autonomic control loop executes four phases, known as MAPE (Monitoring, Analysis, Planning and Execution). The monitoring phase gets events/data from the sensor interface, the analysis phase is executed by the diagnosers, the planning phase determines how to repair a fault detected, and the execution phase sends the commands to the components via the Touch Point.
- **Orchestrating autonomic managers:** coordinates the Local Autonomic Managers.
- **Manual Manager:** creates the human-computer interface for the autonomic managers.
- **Knowledge Sources:** provides access to the knowledge of the middleware.

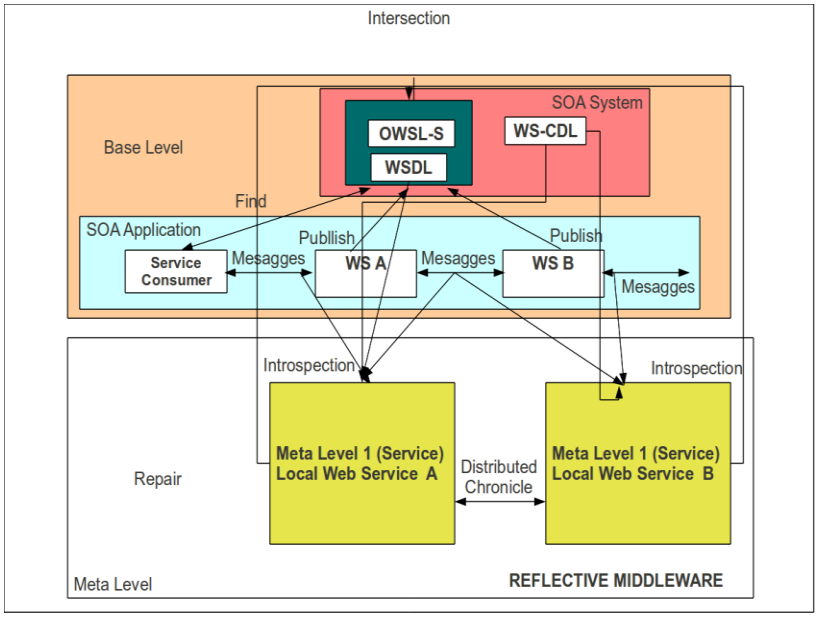

Figure 1. ARMISCOM Architecture

In our case, the *managed resources* and the *touch point* are at the base level; the *autonomic manager*, *the knowledge sources* and *the choreography autonomic manager* are of the meta level (see Fig. 2) [17]. Furthermore, the autonomic manager is composed of three components (diagnoser, repairer and knowledge framework), which are equivalent to the structure MAPE of an autonomic computing architecture. In particular, the diagnoser observes the system and analyzes the failures, and the repairer defines the reparation plans and orders the execution of repair actions.

In ARMISCON each Autonomic Manager works locally (for each service), and through the interaction between autonomic managers is built the diagnosis of failures in the services composition. In particular, the three meta-level modules that composed each autonomic manager are [17]:

- **Diagnoser (Monitor and Analyze):** it inspects the communication services and performs diagnosis. It is invoked by the communication analysis services and has a diagnoser module distributed among the services, to identify the faults (this module is based on chronicles fault patterns).
- **Repairer (Plan and Execute the reparation):** it has mechanisms for the resolution of the fault problems present in the composition of services.

IV. KNOWLEDGE FRAMEWORK COMPONENT

The Knowledge Framework provides the interface to allow the management of knowledge in our middleware. It is composed by (see Fig. 3):

- The SOA System:
  - **Web Services Description Language (WSDL):** It describes how the services can be called, what parameters are expected, and what functionalities are offered.
  - **Web Services Choreography Description Language (WS-CDL):** It describes the Web Services Choreography.
  - **Semantic Markup for Web Services (OWL-S):** It describes semantically the web services using ontologies [8], in order to automate tasks of discovering, invoking, composing, and monitoring of web services.

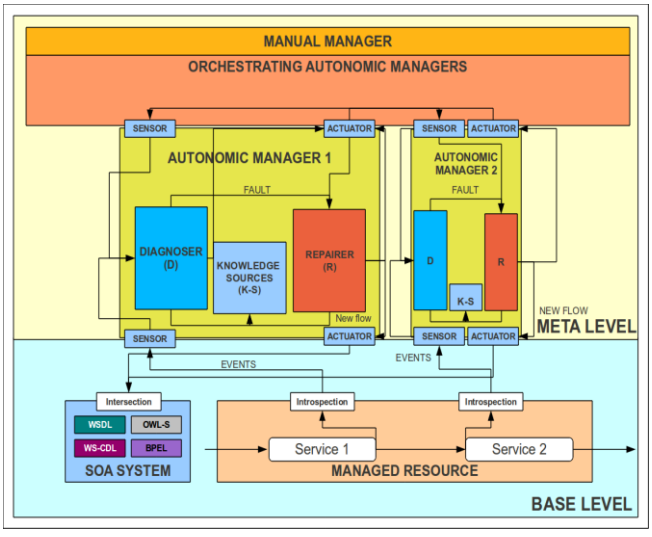

Figure 2. ARMISCON autonomic structure

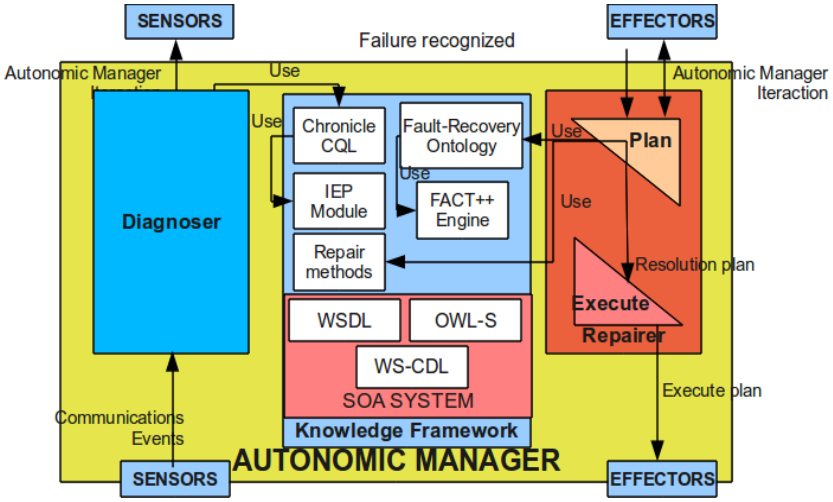

Figure 3. Knowledge Framework components

- **Distributed Chronicles:** It is used mainly by the Diagnoser component. In previous works, we have presented how to represent distributed chronicles, which define the faults, using CQL language [19].
- **Fault-Recovery Ontology:** It is used mainly by the repairer component, in order to define the relationships between the faults and repair methods.
- **Service repair methods:** It is used to store the methods available for the repair service.

A. Distributed Chronicles

In previous work [18, 19], we have designed distributed chronicles in order to specify the different patterns of the faults of the web services. For this, in [18] we have extended the formalism of chronicles, introducing the notion of sub-chronicles, binding events, etc. Furthermore, we have described the process of recognition of our chronicles fully distributed.

Specifically, in [18, 19] we have designed a set of event patterns for recognizing distributed chronicles based on the fault types proposals in [2]. To implement the chronicles we have used the IEP component in OpenESB and the language CQL to define the restrictions between events, in contrast with the tools normally used for recognizing chronicles, as CRS and CarDeCRS. The language CQL allows more expressive by introducing constraints on non-temporal variables [19].

The chronicles are the knowledge about the pattern of behavior of a SOA application when it has a fault. Each chronicle defines a fault type, and it is the knowledge that requires the diagnoser component to detect and diagnose a fault in the application. In [18, 19] are defined the generic patterns (chronicles) for each type of fault defined in [2]. The generic chronicles defined for each fault are:

Physical:

- Unavailable Service Fault

Development:

- Parameters Incompatibility Fault
- Fault due to Interface Might Have Changed
- Fault due to Non-deterministic Actions
- Workflow Inconsistency Fault

Interactions:

- Misunderstood Behaviour Fault (Incorrect Service).
- Response Faults.
- Time-out.
- Misbehaving Execution Flow Fault.
- Incorrect Order.
- Violation of the Service Level Agreement (SLA) and Quality of Service (QoS).

In this way, the knowledge about the behavior of a SOA application with fault is defined using chronicles. Our middleware customizes these generic chronicles according to the specific characteristics of the SOA application supervised

B. Fault-Recovery Ontology

The Fault-Recovery ontology allows correlating faults in the composition of services with available methods for correcting faults in the SOA application. The ontology is the main element of the repairer component, because using it the repairer analyzes the methods of correction of the fault diagnosis by the diagnoser component. The repairer component reasons about the possible methods of corrections of a fault, using the knowledge about that describes in the ontology.

This ontology about the methods the reparation of each fault type in a SOA application is based on the work [2], where they carried out a survey over this topic. The ontology is implemented as a web service that can be accessed by all repairers in our middleware.

Now, we describe the concepts and relationships among them of our ontology. We start describing the concepts of the fault types, then the concepts of the reparation methods, and finally, the generic structure of our ontology where we describe the relationships among the concepts.

C. Concepts about Fault Types in the Web Services Composition

In [2] have described a taxonomy, which classifies the failures in the services composition at three levels: physical, development and interactions. This is the base of our ontology to define the concepts about the fault types.

Physical Faults: Failures are due to the environment where the service (infrastructure) operates and are unrelated to the functionality of the service, causing the service to be considered unavailable (Service or network connection to the service is down). The symptom is that it is not possible to invoke the service (fault due to Unavailable Service).

Development Faults: At the time of conception or development of services and/r composition, may emerge faults that are not considered by the developer of the services and their composition, these types of fault are:

- Parameters Incompatibility Fault: This failure arises when a service is invoked with incorrect values and/or data types of the arguments, with respect to the types and restrictions defined in the WSDL[1] document.
- Fault due to Interface Might Have Changed: The type of data in the interface of some service S_i, which is part of the composition, is modified, so that an incompatibly of parameters is originated when Service S_i is newly invoked. The difference of this fault with respect to parameters incompatibility fault is that the S_i service was previously invoked without failure with the original parameters.
- Fault due to Non-deterministic Actions: This failure occurs when the value of the response of a service is not consistent with the value that should produce the service in the choreography. This kind of failure is extremely rare and is usually because to generate a correct response, the service must previously to invoke another operation in the same service.
- Workflow Inconsistency Fault: In this type of fault the logic in the flow is not correct (Workflow Inconsistency), a service cannot be invoked because its interface does not match the description in the composition. The diagnosis of this type of failure is very complicated, because it is confused with a physical fault (fault due to Unavailable Service).

Interaction Faults: In service composition, interactions occur between services, which can cause faults. In these cases, the types of faults are:

- Misunderstood Behaviour Fault (Incorrect Service): One of the services in the flow of the composition does not produce the expected results. That is not due to that the service does not work properly (it could perform its operations the best possible), but the result is not as expected. To show an example of this, assume that when a service is invoked is expected to return the temperature measured hourly, and the service returns the temperature measured every two hours.
- Response Fault: When the invocation of a service is performed produces a failure in its operation, this may be due to infrastructure problems, authentication or internal logic of the service.
- Time-out: When is described the invocation of a service in the composition, a time period is specified for the response, otherwise a timeout event is generated that allows abort the services composition and avoid other faults in the composition.
- Misbehaving Execution Flow Fault: This fault occurs when a service group or individual service in the composition not yield the expected results in its implementation. They work correctly, but they are not coupled with the other services in the composition, or the result that generate is erroneous within the composition.
- Incorrect Order: Incorrect order failure is because the messages used to interact with the services in the composition arrive in a different order of time than expected.
- Violation of the Service Level Agreement (SLA) and Quality of Service (QoS): Non-functional properties of the services are expressed in terms of QoS and SLA. SLAs are used to describe that capabilities should have the service, and QoS is used to measure the quality of the service based on the response time and quality of the information generated. This fault is generated when the SLA and/or QoS are violated.

D. Concepts about Repair methods in the Services Composition

Once a problem is identified in a services composition, it is necessary to perform a set of actions for the services and/or composition in order to return the system to normal behavior. Thus, different repair methods have been proposed to repair the faults in the composition [22, 23], which are applied depending on the level at which the failure occurs:

Service: These repair methods are applied only at the service level. Some methods of repair of this type are.

- **Retry:** It is applied when a service is temporarily unavailable. In this case, it is suspending the current service execution and the service invocation is retried with known parameters until it becomes available.
- **Substitute a Service:** Is to replace the current service by an equivalent. The compatibility assessment is performed by comparing the interface functionality (WSDL), quality parameters (QoS) and service contracts (SLA).
- **Modify parameters incompatible:** At the time invoking or receiving a service, the message exchanged is incompatible with the definitions of WSDL. The repair involves placing an intermediate service, which is responsible for modifying the input or output messages among the services.
- **Reassign:** This repair method is used when the service does not meet the QoS and/or SLA parameters, the action to take is to reassign the service to a new server to solve the problem. Unlike the substitution of service, this repair method does not seek a new equivalent service, it invokes the same service in a new location.
- **Skip a Service:** Is to jump a service that is part of the composition, which can be running or has not yet been invoked, to continue the execution flow of the composition.

Flow: Is to change the execution flow of the composition.

- **Substitute a Flow:** It is used to faults in some level of the composition. This method consists in replacing part of the flow for equivalent flows, adding new or subtracting services.
- **Redo:** Consists of repeating the invocation of a piece of the flow of the composition, using different parameters taken from previous executions that have worked properly.
- **Alternative behaviors:** is to define an alternative flow to follow the composition in the case of a failure.
- **Skip a flow:** Is to jump a part of the flow in the composition, which can be running or has not yet been invoked, to continue the execution flow of the composition.
- **Change Settings:** Is to change the value of a process variable. This method is used when needed to re- execute a portion of the flow, but using different values of the process variables.

E. Relationships in our Fault-Recovery Ontology

The design of our ontology contains two classes, called Fault and Repair Strategies (see Fig. 4), which represent the concepts of failure and repair methods described in sections 4.A and 4.B. Thus, the Fault class has a property called Has_repair_method, which allows us to assign elements of the class Repair Strategies to each type of fault. In this way are matched the failures in the services composition with the mechanisms to solve the faults. It is a superclass of the classes Physical, Development and Interaction. Also, the class Repair Strategies has a property, called Solve_fault, which performs the inverse operation to Has_repair_method, and it is a superclass of the classes Service and Flow.

The individual instances developed in our ontology are shown in Table I. Repair methods for each failure shown in Table I should be taken as a possible set of actions to run to solve the fault, this selection should be done sequentially among the methods available on site. That is, the repair component must try to solve the fault with the first action available (best case), and if with this one is not possible to solve the problem, it continues sequentially with the next action, until repair the fault or reach the last option (worst case). For example, for the failure of unavailable service, the first action is to try to place the service again available (redo service (best case)), in case it cannot be performed, the second action is to try reassign service on another site, if it cannot solve the problem, it is necessary to try the service substitution by an equivalent, and so until repair the fault or test the last action (skipFlow service (worst case)).

We have implemented our ontology using the protégé[1] tool, which is based on the Web Ontology Language (OWL). Subsequently, the repair component of ARMISCOM invokes a service, which reasons and makes inferences about the repair mechanisms according to the failures present in the composition, using our ontology and the inference motor FACT++[2] of protegé.

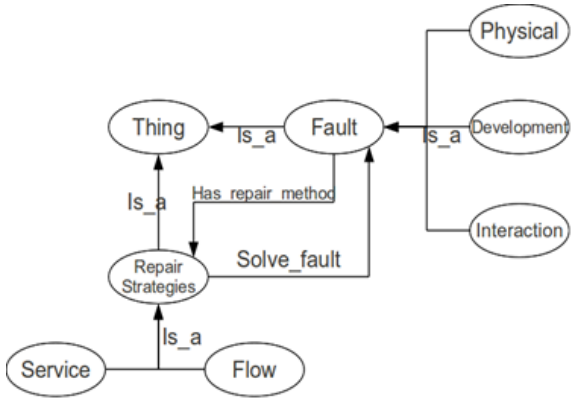

Figure 4. Fault-Recovery ontology structure

Thus, this part of the Knowledge component is implemented by a web service which uses our ontology and the reasoner FACT++. The relationships between the concepts of faults and repair methods in the ontology generated in protegé are shown in Fig. 5

F. Setting the Services of repair methods (Metadata about repair methods)

As shown in the previously proposed ontology, a fault in the composition may have different repair methods. Although some resolution mechanisms can be setting in real time as redo, parametersUpdate, skip service, etc., others need to be previously setting. For example, the method "substitute a service" needs previously to identify the equivalent services, using like knowledge base the SOA system (UDDI, WSDL, OWL-S), because search equivalent services takes some time (it cannot be implemented in real time). Additionally, not all correction mechanisms may be used in some cases/sites, then it is necessary to define a knowledge base that allows ARMISCOM chooses the reparation mechanisms for each case/site.

In these cases, it is necessary to define a mechanism that allows the middleware has stored alternative flows for its repair mechanisms, in order to provide a consistent and quick reparation of a SOA application. Distributed repairs in ARMISCOM are continually looking for equivalent services to replace the service that is responsible when there is a malfunction. Get equivalent services often is not an easy task, and in many cases it is necessary to modify the execution flow of the SOA application (add or remove services). Each repair component continuously updates the metadata with new services and equivalent flows. Because in ARMISCOM the component responsible for performing failure analysis conceives the composition as a stream of events, it is necessary to expand the representation of sub-flows as a sequence of events. In this way, a SOA application can be viewed as a sequence of events E, which can be decomposed into sub-regions or sub-flows R_i of events Eac_i:

[1] Protégé is a free, open source ontology editor. It provides a graphic user interface to define ontologies. This application is written in Java.

2 FaCT++ is a tableaux-based reasoner for expressive Description Logics (DL) developed by the University of Manchester, It covers OWL and OWL2 languages.

TABLE I
INDIVIDUAL INSTANCES IN OUR FAULT-RECOVERY ONTOLOGY

SubClass	individual instance	Has_repair_method
physical	unavailable	• redo • reassign • substitute • substituteflow • skipService • skipFlow
development	parameterIncompatibility	• CompleteMissingParameters • substitute • substituteflow • skipService • skipFlow
	interfaceMightHaveChanged	• CompleteMissingParameters • substitute • substituteflow • skipService • skipFlow
	DueToNonDeterministicActions	• parametersUpdate • substitute • substituteflow • skipService • skipFlow
	workflowinconsistency	• substituteflow • skipFlow
interaction	misunderstoodBehaviourFault	• parametersUpdate • substituteflow • skipFlow
	responsefault	• substitute • substituteflow • skipService • skipFlow
	timeout	• reasign • retry • substitute • substituteflow • skipService • skipFlow
	misbehavingExecutionFlow	• redo • substituteflow • skipFlow
	IncorrectOrder	• substituteflow • skipFlow
	QualityOfService	• reasign • substitute • substituteflow
	ServiceLevelAgreement	• parametersUpdate • reasign • substitute • substituteflow

$$Application(E) = UNION_{i=1,\,n}(R_i(Eac_i)) \quad (1)$$

Where,

- Eac_i are a set of events, such that $Eac_i = \{E_k,,..., E_l\}$ occur in the region i.
- UNION is a predicate that defines the union of distributed events (Eac_i) in the n regions.

Suppose the SOA application shown in Fig 6. This application can be decomposed into regions associated with event services, such that:

$$Application = UNION\{R_1(E_1), R_2(E_2, E_3, E_9), R_3(E_4, E_5), R_4(E_6, E_7), R_5(E_{10}, E_{11}), R_6(E_8, E_{12}, E_{13})\} \mid \forall k,m < 13 \; y \; \forall i,j < 6,\; i \neq j,\; E_k \in R_i \; y \; E_m \in R_j, \text{ then, } R_i(E_k) \cap R_j(E_m) = \emptyset \quad (2)$$

Based on regions R_1, R_2, R_3, R_4, R_5, R_6, it is possible to find equivalent regions R'_1, R'_2, R'_3, R'_4, R'_5, R'_6. Thus, to manage the equivalent regions of the SOA application within ARMISCOM, we need to define a metadata to store repair mechanisms in each case. Repairing a flow of the composition is to find an equivalent region that allows mapping the initial event E_0 and final E_F, that is, one must know the stored equivalent regions related to each repair mechanism.

For that, in ARMISCOM is defined a metadata for each service with the repair methods that can be used in equivalent region (see Table II). In Table II, each attribute is defined as:

- **Weight:** Represents the order in which methods should be extracted, it can be defined based on some kind of optimization.
- **RepairMethod:** Represents the method of reparation available.
- **Flow:** defines the sequence of events (flow) which are affected during the reparation.
- **Flow_init:** Represents the first event on the services composition, in the which should begin the reparation.
- **Flow_end:** Represents the last event on the services composition, in the which should be completed the reparation.

With this metadata, ARMISCOM can define the repair methods available for each case/site. The metadata works as follows: suppose that is necessary to implement the repair method "substitute flow" from event 5 until event 9, then we need to perform the next search: WHERE RepairMethod = "substitute flow" AND Flow_init = 5 AND Flow_end = 9, return data BY Weight. Additionally, because the query could not find any method for the desired flow to modify, the repair component could perform a new search based on a new flow (Eg: Flow_init = 4 and Flow_end = 9 and the same method "substitute flow" describe.

TABLE II
METADATA FOR EACH SERVICE WITH THE REPAIR METHODS

Reparation methods available in a site (service)				
Weight	**RepairMethod**	**Flow**	**Flow_init**	**Flow_end**

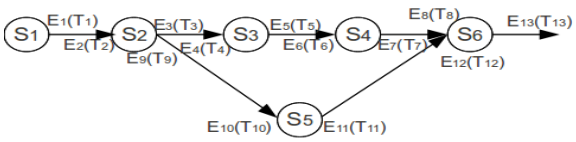

Figure 6. SOA application decomposed into events region

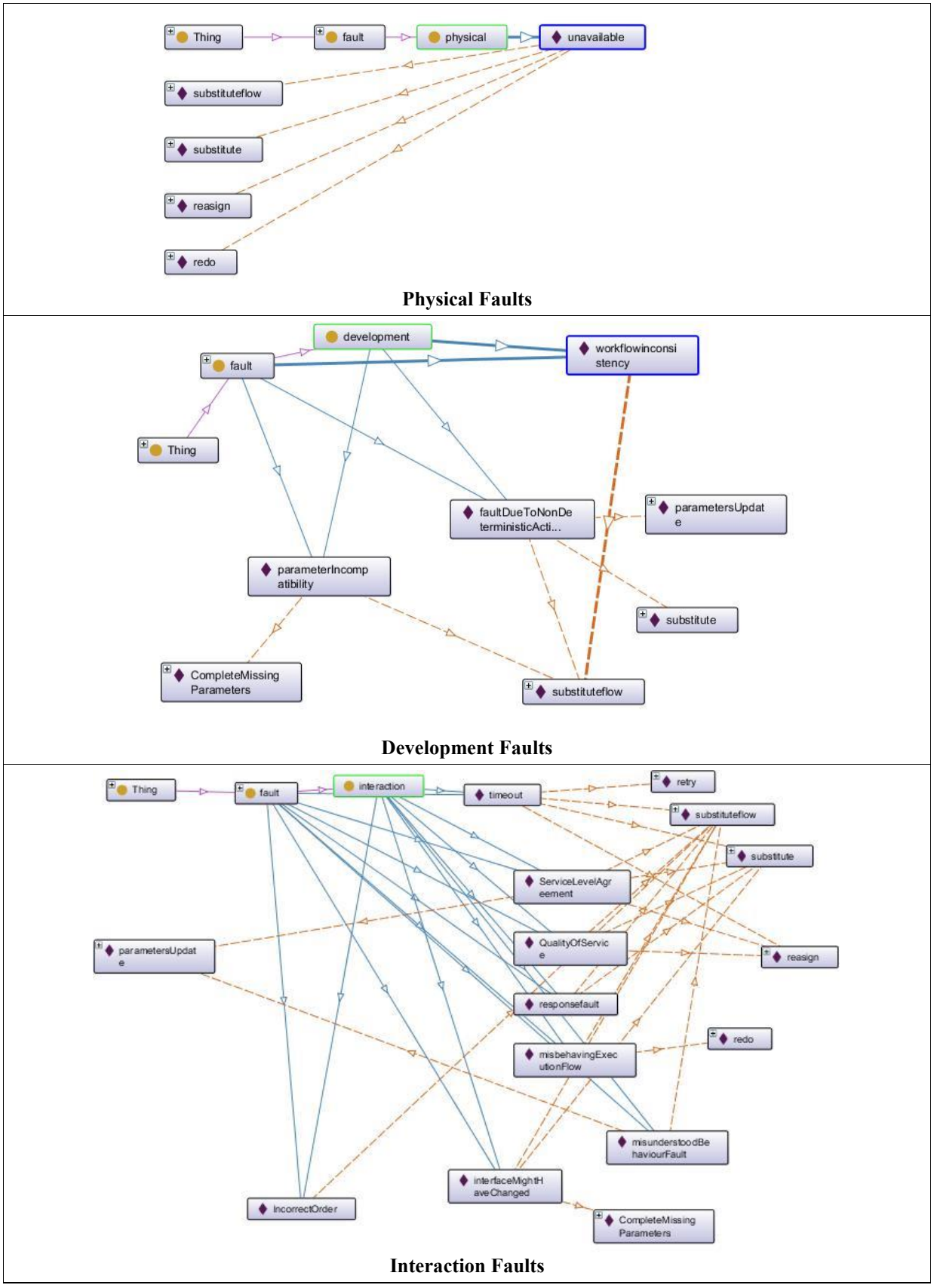

Figure 5. Relationships among the concepts in our Fault-Recovery ontology.

V. Case Study

In this test case we will use a common example of e-commerce SOA implementation (see Fig. 7), which comprises three business processes (which will constitute our services):

- **Shop:** it is the place where users purchase products.
- **Supplier:** it offers products to the shop, but needs to verify their availability before to response.
- **Warehouse:** it is the place where the products are stored by the providers. This service has a service level agreement (SLA)[3] with Supplier, which is that at least one product from the list should be returned[4]. It can interact with other warehouses of the company, in order to search products. In this way, it can answer with at least one product, when it has not in the local warehouse.

Now, we describe a classical behavior of this application:

(1) **SuppListOut:** Shop provides the list of products required to the supplier.
(2) **SuppItemIn:** Supplier checks its deposit invoking the Warehouse process.
(3) **SuppItemOut:** Warehouse provides the list of products in the deposit to the Supplier.
(4) **SuppListIn:** The Supplier informs the products that can provide to the Shop.

A. Some elements of the knowledge component of ARMISCOM in this case

In the case of chronicles, Fig. 8 and Table III define the distribution of the events among the diagnosers (sites) of the composition, which is a generic chronicle for this application (connecting all events that may occur in it). With this generic chronicle, can be built the specific chronicles to detect each abnormal situation.

Based on the patterns of the generic chronicles for the different types of faults of a SOA application proposed in [18, 19], the knowledge component builds the specific distributed chronicle for each fault: Quality of Service, Timeout, etc. One example of one of these chronicles is shown in Table IV in the cases of Timeout and Quality of Service.

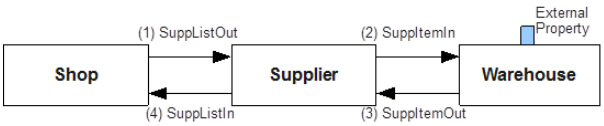

Figure 7. Example of choreography (e-commerce).

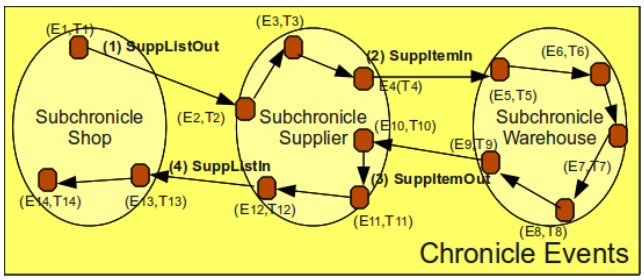

Figure 8. Sequence event divided by diagnoser in E-Commerce case.

TABLE III
EVENT DESCRIPTION DIVIDED BY DIAGNOSER IN E-COMMERCE CASE

Shop	Supplier	Warehouse
• **E_1:** Shop sends product orders to the Supplier. • **E_{13}:** Shop receives the list of products. • **E_{14}:** Shop makes products payment.	• **E_2:** Supplier receives product orders • **E_3:** Supplier checks the products in the catalog. • **E_4:** Supplier provides product orders to Warehouse for the products that it has not. • **E_{10}:** Supplier receives the response of the products. • **E_{11}:** Supplier makes the invoice. • **E_{12}:** Supplier responds to shop with products shipped.	• **E_5:** Warehouse receives the request of the Supplier. • **E_6:** Warehouse searches products (maybe it invokes other warehouses). • **E_7:** Warehouse updates inventory. • **E_8:** Warehouse packs and ships products to the buyer. • **E_9:** Warehouse provides the answer about the list of products in the deposit to the Supplier.

Tmeout:

- **Subchronicle Supplier:**
  - **Input:**
    - ***E4*** is an event that is maintained by ***15000 ms*** and ***ENOEVENT:*** is a stream produced by the no response from the warehouse. Both **E4** as **ENOEVENT** have no temporal attributes ***id*** (is an identifier used to ensure that the events corresponding to the invocation of the application itself), ***time*** (generated when the event occurs) and ***lp*** (products list) .
  - **Constraint:**
    - The events should have the same id, and the time difference between ***ENOEVENT*** and ***E4*** must be ***5000 ms***.
  - **Output**
    - Emit a bidding event call ***EBTimeout*** to Warehouse diagnoser**:**
  - **Subchronicle warehouse:**
    - **Input:**
      - ***E5***, ***E6*** and ***E7*** are events maintained by ***15010***, ***15008*** and ***15006*** respectively; ***EBTimeout*** is a stream. All have the same attributes id, time and lp, as in Subchronicle Supplier.
    - **Constraint:**
      - The events must be the same ***id*** and the arrival sequence of the events is established.
    - **Output:**
      - Emit an event to repair, with fault information. To this, we have added additional information to the event, to tell the repairer the name and type (timeout) of the fault, and the affected flow (flow_init = 5 and flow_end = 9, the affected flow are a five services).

[3] SLA is a contract between the service consumer and service provider and define the level of service

[4] This SLA define how message delivery is guaranteed, the Warehouse delivery messages in the proper order (least one product in order)

TABLE IV
DISTRIBUTED CHRONICLES FOR TIMEOUT FAULT AND QUALITYOFSERVICE IN CQL

Distributed Chronicle: Timeout

```
Subchronicle Supplier Timeout {
SELECT
    ISTREAM(
     id => E4.id,
     event => 'EBTimeout',
     time => E10.time,
     lpsupplier => E10.lp,
     lp => E4.lp,
     to => 'Diagnoser warehouse',
   )
FROM
   E4[15000],
   ENOEVENT[now]
WHERE
   ENOEVENT.time >= E4.time + 5000 AND
   ENOEVENT.id = E4.id
}
```

```
Subchronicle Warehouse Timeout {
SELECT
   ISTREAM(
     id => E5.id,
     fault => 'timeout',
     faulttype => 'N/A',
     time => E4.time,
     lp => E4.lp,
     flow_init => 5,
     flow_end => 9,
     to => 'Repair warehouse',
   )
FROM
   E5[15010],
   E6[15008],
   E7[15006],
   EBTimeout[now]
WHERE
   E6.time > E5.time AND
   E7.time – E6.time > 4 AND
   E6.id = E5.id AND
   E7.id = E6.id AND
   EBTimeout..id = E7.id
}
```

Distributed Chronicle: QualityOfService: Delay

```
Subchronicle Supplier Delay0 {
SELECT
   ISTREAM(
     id => E4.id,
     event => 'EBDelay',
     time => E10.time,
     lp => E4.lp,
     to => 'Diagnoser supplier',
   )
FROM
   E4[5500],
   E10[now]
WHERE
   E10.time - E4.time >= 2000 AND
   E10.time - E4.time < 5000 AND
   E10.id = E4.id AND
}
```

```
Subchronicle Supplier Delay1{
SELECT
   ISTREAM(
     id => EBDELAY1.id,
     fault => 'QualityOfService',
     faulttype => 'Delay',
     time => E10.time,
     lp => E4.lp,
     flow_init => 8,
     flow_end => 8,
     to => 'Repair supplier',
   )
FROM
   EBDELAY[15500],
   EBDELAY1[now],
WHERE
   count(EBDELAY.id) + 1 > 2 AND
    EBDELAY.id <>  EBDELAY1.id
}
```

Quality of Service (Delay):

- **Subchronicle Supplier 1:**
  - **Input:**
    - E_4 is an event that is maintained by ***55000 ms*** and E_{10} is a stream. They have attributes ***id***, ***time*** and ***lp***.
  - **Constraints:**
    - The difference in the time of events E_4 and E_{10} should be between ***2000*** and ***5000 ms***.
  - **Output:**
    - Emit a bidding event, called EB_{Delay}, to Supplier diagnoser.

- **Subchronicle Supplier 2:**
  - **Input:**
    - EB_{Delay} is an event maintained by ***15500 ms*** and EB_{Delay1} is a stream, both have attributes ***id***, time and lp.
  - **Constraint:**
    - The amount of received events must be greater than ***2***
  - **Output:**
    - Emit an event to repair in supplier, with fault information. To this, we have added additional information to the event, to tell the repairer the name and type of the fault (name = Quality Of Service type = Delay), and the affected flow (flow_init = 8 and flow_end = 8, the affected flow is a unique service).

Additionally, the service of the repair methods available at each site, and their metadata, are shown in Table V.

Thus, the supplier only has a repair mechanism (substituteflow) affecting the flow from the event 5 until the event 9 (repair E5, E6, E7, E8 and E9 operations), On the contrary, warehouse has four repair mechanisms: parametersUpdate (repair E6 operation), CompleteMissingParameter (repair E5 operation), substituteflow (repair E8 operation) and substituteflow (repair E7 operation).

B. Testing the e-commerce application using the Knowledge Component

To verify the operation of component of knowledge of ARMISCOM, we implement the application of E-commerce in OpenESB and connected the distributed diagnoser and repair modules. At the Warehouse service we have added one additional operation to easily induce delay faults and to verify its full operation:

setTuneDelay: Used to induce delay time in the warehouse service (initial delay is 0 ms, no delay). Thus, three invocations of the application are performed (id = {1, 2, 3}) where TuneDelay is setting with a delay of 3000 ms (induces multiple delay fault). Subsequently is invoked again the warehouse service (id = 4) with a TuneDelay of 6000ms what would cause a timeout in e-commerce application. The results are shown in Table VI.

TABLE V
AVAILABLE METHODS TO REPAIR E-COMMERCE APPLICATION

reparation methods available in Supplier				
Weight	**RepairMethod**	**Flow**	**Flow_init**	**Flow_end**
1	substituteflow	E6, E7, E8, E9	5	9
reparation methods available in Warehouse				
Weight	**RepairMethod**	**Flow**	**Flow_init**	**Flow_end**
1	parametersUpdate	E6	6	6
1	CompleteMissingParameter	E5	5	5
1	substituteflow	E8	8	8
1	substituteflow	E7	7	7

TABLE VI
KNOWLEDGE SOURCE USED IN QUALITY OF SERVICE (DELAY) AND TIMEOUT FAULTS

Fault	**Distributed Chronicle Diagnosis Response**	**Fault-Recovery Ontology Response**	**Service repair Selection Response**
Quality Of Service (Delay)	<?xml version="1.0" encoding="utf-8"?> <msgns:StreamOutput4_MsgObj xmlns:msgns="supplierChronicle_iep"> <id>3</id> <fault>QualityOfService</fault> <faulttype>Delay</faulttype> <time>1408573740066</time> <lp>10</lp> <flow_init>8</flow_init> <flow_end>8</flow_end> <Timestamp>2014-08-20T17:59:05.927-04:30</Timestamp> </msgns:StreamOutput4_MsgObj>	<?xml version="1.0" encoding="utf-8"?> <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope"> <S:body> <ns2:getRepairMethodResponse xmlns:ns2="http://ws/"> <return>reassign;substitute;substituteFlow</return> </S:body> </S:Envelope>	<?xml version="1.0" encoding="utf-8"?> <msgrepair:supplier> <methodrepair>substituteflow</methodrepair> <flow_init>8</flow_init> <flow_end>8</flow_end> </msgrepair:supplier>
Timeout	<?xml version="1.0" encoding="utf-8"?> <msgns:StreamOutput2_MsgObj xmlns:msgns="warehouseChronicle_iep"> <id>4</id> <fault>timeout</fault> <faulttype>N/A</faulttype> <time>1408573625710</time> <lp>2</lp> <flow_init>5</flow_init> <flow_end>9</flow_end> <Timestamp>2014-08-20T17:57:06.025-04:30</Timestamp> </msgns:StreamOutput2_MsgObj>	<?xml version="1.0" encoding="utf-8"?> <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope"> <S:body> <ns2:getRepairMethodResponse xmlns:ns2="http://ws/"> <return>reassign;retrysubstitute;substituteflow;skipService;skipFlow</return> </S:body> </S:Envelope>	<?xml version="1.0" encoding="utf-8"?> <msgrepair:supplier> <methodrepair>substituteflow</methodrepair> <flow_init>5</flow_init> <flow_end>9</flow_end> </msgrepair:supplier>

As shown in Table V, ARMISCOM was able to diagnose and correct Quality of Service (Delay) and Timeout faults. In the case of Quality of Service (Delay), the supplier diagnoser recognizes chronicle and emits the event to its repairer (fault: QualityOfService, fault type: Delay, flow_init = 8 and flow_end = 8, see first column). With this information the repairer performs inference in the Fault-Recovery ontology for the QualityOfService fault, and returns the possible solution methods to be implemented to correct the fault (reassign, substitute and substituteFlow, see second column). Then, the repair performs the search in metadata: first it searches method reassign, after substitute, and because they are not implemented, subsequently seeks substituteFlow with flow_init = 8 and flow_end = 8 (taken from Distributed Chronicle Diagnosis Response). This one is available to be applied like method to solve the fault. The diagnosis and correction of the Timeout fault is similar. First, the Warehouse Diagnoser recognizes the chronicle and emits the event to its repairer (fault: timeout, fault type: N/A, flow_init = 5 and flow_end = 9, see first column). The repair carries out an inference about the fault in the Fault-Recovery ontology, and returns the possible methods to implement (reassign, retrysubstitute, substituteflow, skipService and skipFlow). Finally, it performs a search in the metadata of the Warehouse repairer to find possible repair mechanisms to implement, the repair mechanism "substituteflow", for flow: flow_init = 5 and flow_end = 9, is the only one available.

C. Results Analysis

In the case study, we observe how the knowledge component of ARMISCOM uses hybrid knowledge to manage the different aspects necessary to guarantee the fault tolerance of a SOA application. The different patterns of distributed chronicle are used to diagnose the failures (in this case, we have shown the Quality Of Service (Delay) and Timeout chronicles). When a distributed chronicle is recognized the diagnoser produces a file with the diagnosis, which is read by the repair component. This component uses the Fault-Recovery Ontology to reason about the repair methods that could be used to solve the fault. Finally, with the identification of the part that has been affected (event_init and event_end) and the repair mechanisms stored in the metadata, ARMISCOM can get the best available method to solve the fault in real time.

Our extension of the formalism of chronicles, facilitates the interactions between local diagnosers, without need of a coordinator to manage their interactions. This represents a remarkable improvement in communication and scalability level, with respect to previous studies [13, 14, 15, 16]. In addition, its implementation is very natural in the case study (a recognizer by service).

Some works store subflows modeling them as a set of services that are interconnected with each other, using Petri nets or graphs connections [3, 4, 6, 12, 22, 24, 25, 26, 27, 28]. Additionally, they replace sub-flows in the composition of services, have architectures that allow them to previously find alternate sub-flows, to respond to faults present in the composition in real time. Thus, the mechanisms consist of modeling a SOA application as a graph or path, which can be decomposed into sub-graphs, and achieve equivalent flows based on a similarity criterion, according to the functional and non-functional (e.g. QoS).

In this work, flows have been modeled as events with time constraints, to be in line with the chronicle paradigm. Additionally, the metadata can store the information that characterizes the regions of the events (Initial Flow Event (Event_init), sequence of events that compose it (Transition), Final Event Flow (Event_end)), which defines the region where must be applied the repair strategy (RepairMethod), and determines the equivalent regions. All this information can be used to select the best option, in order to be effective when a service must be replaced.

Additionally, we have designed a repair module that allows us to infer the repair strategies for failures in the services composition, taking into account context information based in the fault and in the flow composition problem, which is performed at runtime. In previous words [9, 10] have correlated recovery actions with fault type, in our case we use a fault-recovery ontology to correlate the faults with the recovery actions, which was implemented as a web service using BC Sun Java EE SE, to encapsulate the JAVA language as a service, and the inference engine FACT++. The various queries performed at service ontology for each failure showed the expected response (repair methods to use) in the reparations. This ontology can increase (e.g. using ontological learning approaches) to include new faults, reparation mechanisms, etc. Also, the metadata provides to ARMISCOM multiple recovery plans, to address the flow fault in the composition of web services. The case study showed how to store different repair mechanisms and to make the request to the meta-data, in order to find the mechanisms best suited to the part affected (which failed). In this way, ARMISCOM can be customized very easily, because can deduce the appropriate repair method to be used for each case.

VI. Conclusions

We have proposed a reflective middleware architecture for autonomic management of service-oriented applications [17]. ARMISCOM is fully distributed through the services of the SOA application, it is instanced in each service, for both the diagnosis and the reparation of faults of services and of compositions. In order to support this architecture, in this paper, we have designed the knowledge management component of our middleware. This Knowledge is composed of the information from SOA system, of Distributed chronicles which describe the behavior of a SOA application with failures, the distributed metadata which describes the repair methods, and of a Fault-Recovery Ontology.

In the case of distributed chronicles, previously, in [18], we have extended the formalism of chronicles, with the definition of the notion of sub-chronicles, binding events, among others. Our extension contrasts with the semi-centralized and decentralized chronicle approaches that have been developed previously.

Additionally, chronicles make possible to identify the parts affected by the faults, adding new attributes to the events as fault name, fault type, part of the flow affected by the failure (flow_init and flow_end). With this information, in this paper,

we have described as ARMISCOM determines the equivalent regions, which are sub-flows as events with time constraints. In this way, ARMISCOM can characterize regions with fails to be replaced, which defines the region where must be applied the repair strategy (RepairMethod),

In the case of the Fault-Recovery Ontology component, it has been implemented as a web service, allowing correlated faults present in the composition with repair mechanisms using an inference motor. The Fault-Recovery Ontology component has been developed as an ontology composed of super-classes, classes, properties and individuals using OWL language, which describe a taxonomy about the mechanisms of reparation of faults for a SOA application. This ontology can be enriched in the future to allow inferences about the more complex situations, using functional and non-functional properties of services.

Finally, we have proposed a metadata about each repair methods available at each site, which must be used by the repair component. Using this metadata the repairer deduces the appropriate repair method for each case. Metadata provides representation of multiple recovery plans available at different instances of flows (web services) of the composition of services, using the concept of equivalent regions, which allows to calculate the suitable plan to implement in the case of a fault.

Our middleware requires a knowledge component which manages hybrid knowledge, in order to properly infer the portion of the flow that has failed and find the closest resolution mechanism. This architecture for autonomic management of service-oriented applications is based on hybrid knowledge, according to the needs of each MAPE component. The utilization of the hybrid knowledge (different sources of knowledge) defined in this paper, is one of the advantages of our approach. Additionally, the component of the distributed Knowledge representation designed in this paper, allow the self-healing web service composition fully distributed, representing another significant improvement, in order to reduce the large exchange of messages and to minimize the calculation required in the diagnosis and the reparation, which are the main problems of the centralized approaches [3, 4, 6, 12, 22, 25, 26, 27, 28].

Some improvements are possible. For example, the metadata are defined by an expert. However, this task could be delegated to another component that automatically build it. An example is to use another ontology to infer services and flow equivalences, this would work as a robot that is continuously running and updating the metadata, which can be enriched using the weight field for indicating the degree of equivalence. Also, the ontology can be extended to describe features that allow to infer the services of reparation more exactly

REFERENCES

[1] Czajkowski, K., Fitzgerald, S., Foster, I., and Kesselman, C.: "Grid Information Services for Distributed Resource Sharing". *10th IEEE International Symposium on High Performance Distributed Computing*, pp. 181--184. 2001.

[2] Chan, K., Bishop, J., Steyny, J.,Baresi, L., and Guinea, S.:, "A Fault Taxonomy for Web Service Composition", *Service-Oriented Computing Workshop*, pp. 363-375, 2007.

[3] Huang, G., Liu, X., and Mei, H.: "SOAR: Towards Dependable Service-Oriented Architecture via Reflective Middleware". *International Journal of Simulation and Process Modelling*, vol. 3, no. 1/2, pp. 55-65, 2007.

[4] R. Halima, E. Fki, K. Drira and M. Jmaiel, "Experiments results and large scale measurement data for web services performance assessment". *IEEE Symposium on Computers and Communications*, pp. 83-88, 2009.

[5] WS-Diamond project, "WS-Diamond, IST-516933, Deliverable D4.3, Specification of diagnosis algorithms for Web Services – phase 2", http://wsdiamond.di.unito.it/.

[6] Poonguzhali, S.,, Sunitha, R., and Aghila, G.: "Self-Healing in Dynamic Web Service Composition". *International Journal on Computer Science and Engineering*, vol. 3, no. 5. pp. 2054-2060, 2011.

[7] IBM Corporation. "An architectural blueprint for autonomic computing". Autonomic Computing", Fourth Edition, http://www.ginkgo-networks.com/IMG/pdf/AC_Blueprint_White_Paper_V7.pdf, 2006.

[8] Chiribuca, D., Hunyadi, D. and Popa, E.: "The Educational Semantic Web", *8th WSEAS International Conference on Applied Informatics and Communications*, pp. 314-319, 2008.

[9] Fugini, M.G., Mussi, E.: Recovery of Faulty Web Applications through Service Discovery. *32nd International Conference on Very Large Databases*, pp. 67-80, 2006.

[10] Ardagna, D., Cappiello, C., Fugini, M., Mussi, E., Pernici, B., andPlebani, P.: Faults and recovery actions for self-healing web services. *15th Int. World Wide Web Conf.*, 2006.

[11] Sherif, A.; and Amir, Z.: Towards autonomic web services: achieving self-healing using web services. *2005 Workshop on Design and evolution of autonomic application software*, Pages 1 – 5, 2005.

[12] Poonguzhali1, S.; JerlinRubini, L.; Divya, S.: "A Self-Healing Approach for Service Unavailability in Dynamic Web Service Composition". *International Journal of Computer Science and Information Technologies*, vol. 5 Issue 3, p 4381, 2014.

[13] WS-Diamond: WS-Diamond, IST-516933, Deliverable D4.3, Specification of diagnosis algorithms for Web Services – phase 3. Version 0.5, 2008.

[14] Cordier, M.O., Krivine, J., Laborie, P., Thi´ baux, S.: "Alarm processing and reconfiguration in power distribution systems". *IEA-AIE'98*. pp. 230–240, 1998.

[15] Cordier, M.O., Dousson, C.: "Alarm driven monitoring based on chronicles". *Safeprocess'200*0. Pp 286–291, 2000.

[16] Quiniou, R., Cordier, M.O., Carrault, G., Wang, F.: "Application of ilp to cardiac arrhythmia characterization for chronicle recognition". *ILP'2001.* pp. 220–227, 2001.

[17] Vizcarrondo, J., Aguilar, J., Exposito, E., Subias, A.: "ARMISCOM: Autonomic Reflective MIddleware for management Service COMposition". *4th Global Information Infrastructure and Networking Symposium (GIIS 2012)*, IEEE Communication Society, 2012.

[18] Vizcarrondo, J., Aguilar, J., Exposito, E., Subias, A.: "Crónicas Distribuidas para el Reconocimiento de Fallas", *Revista Ciencia e Ingeniería.* vol. 36, no. 2, pp. 73-84, 2015.

[19] Vizcarrondo, J., Aguilar, J., Exposito, E., Subias, A.: "Building Distributed Chronicles for Fault Diagnostic in Distributed Systems using Continuous Query Language (CQL)", *International Journal of Engineering Development and Research (IJEDR),* vol.3, no. 1, pp.131-144, 2015

[20] Aguilar, J. "An artificial immune system for fault detection", *Intl. Conf. on Industrial, Engineering and other Applications of Applied Intelligent Systems*, pp. 219-228, 2004.

[21] Aguilar, J., Hernández, M. "Fault tolerance protocols for parallel programs based on tasks replication", *8th Intl Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 397-404, 2000.

[22] Ardissono L., Console L., Goy A., Petrone G., Picardi G., Segnan M, "Enhancing Web Services with Diagnostic Capabilities". *Third European Conference on Web Services*, pp. 182-191, 2005.

[23] Fugini ,M. Mussi G: "Recovery of Faulty Web Applications through Service Discovery". *32nd International Conference on Very Large Databases*, pp. 67-80, 2006.

[24] WS-Diamond: WS-Diamond, IST-516933, Deliverable D5.1, Characterization of diagnosability and repairability for self-healing Web Services, 2005.

[25] Feng X., Wang H., Wu Q., Zhou B, "An adaptive algorithm for failure recovery during dynamic service composition," in *Pattern Recognition*

and Machine Intelligence (A. Ghosh, R. De, and S. Pal, Eds). Springer Berlin / Heidelberg, vol. 4815, pp. 41-48, 2007.

[26] Feng X., Wu Q., Wang H., Ren Y., Guo C, "ZebraX: A model for service composition with multiple QoS constraints", In *Advances in Grid and Pervasive Computing* (C. Cerin, K.-C. Li, Eds.), Springer Berlin/Heidelberg, vol. 4459, pp 614-626, 2007.

[27] Canfora G., Di Penta M.., Esposito R., Villani M, "A framework for QoS-aware binding and re-binding of composite web services", *Journal of Systems and Software*, vol. 81, pp. 1754-1769, October 2008.

[28] Saboohi P., Amini A., Abolhassani H., "Failure recovery of composite semantic web services using subgraph replacement,, *International Conference on Computer and Communication Engineering (ICCCE)*, pp. 489-493, 2008.

Juan Vizcarrondo is System Engineer, and obtained a Msc in Computer Science at the Universidad de los Andes, Mérida-Venezuela, and a PhD in Computer Science at the Universidad de los Andes. He works at the Cenditel since 2007.

Jose Aguilar is a System Engineer graduated in 1987 from the Universidad de los Andes, Merida, Venezuela. M. Sc. degree in Computer Sciences in 1991 from the University Paul Sabatier-Toulouse-France. Ph. D degree in Computer Sciences in 1995 from the University Rene Descartes-Paris-France.. He completed post-doctorate studies at the University of Houston, researcher at the Microcomputer and Distributed Systems Center (CEMISID) at the same university. Member of the Mérida Science Academy and the International Technical Committee of the IEEE-CIS on Artificial Neural Network.

Ernesto Exposito earned his engineer degree in computer science from the "Universidad Centro-occidental Lisandro Alvarado" (Venezuela, 1994). He earned his PhD in “Informatique et Télécommunications” from the Institut National Polytechnique de Toulouse (France, 2003). He is Professor in computer sciences at the Institut National des Sciences Appliquées (INSA) of Toulouse.

Audine Subias received a PhD degree in 1995 and a M.S.degree in 1992 in Informatique Industrielle, both from Paul Sabatier University, in Toulouse, France. Since 1997 she is Associate Professor in control and discrete event systems at the Institut National des Sciences Appliquées (INSA) of Toulouse.

MiR-EO: Middleware Reflexivo para la Emergencia Ontológica en Ambientes Inteligentes

MiR-EO: Reflective Middleware for Ontological Emergency in Intelligent Environments

Mendonça Maribel, Aguilar Jose, Perozo Niriaska

***Resumen*— En un Ambiente Inteligente (AmI), los dispositivos que participan deben intercambiar conocimiento permanentemente, para lo cual deben entenderse y manejar un lenguaje común, para el logro de la interoperabilidad semántica. Las ontologías en un AmI constituyen una herramienta ideal para ello, posibilitando la comunicación entre los objetos inteligentes que forman parte del ambiente. Estas ontologías deben ser distribuidas, heterogéneas y dinámicas ya que deben adaptarse a los cambios, necesidades y servicios del AmI. Este artículo propone la implementación de un middleware que permite la emergencia ontológica, con el fin de gestionar todo el conocimiento que se puede generar en un AmI. El middleware, llamado MiR-EO, se implementa como un middleware reflexivo, que maneja su propio marco ontológico, conformado por meta-ontologías que modelan los elementos que deben contener las ontologías de un AmI, y posibilitan el proceso de emergencia ontológica.**

***Palabras Claves*— Emergencia ontológica, middleware reflexivo, ambientes inteligentes.**

***Abstract*— In a Smart Environment (AmI), the devices that participate must exchange knowledge permanently, for which they must understand and manage a common language. The ontologies in an AmI are an ideal tool for this, making possible the communication between the intelligent objects that are part of the environment. These ontologies must be distributed, heterogeneous and dynamic, since they must adapt to the changes, needs and services of the AmI. This article proposes the implementation of a middleware that allows the ontological emergence, to manage all the knowledge that can be generated in an AmI. This middleware, called MiR-EO, is implemented as a reflective middleware, which manages its own ontological framework, made up of meta-ontologies that model the elements that must contain the ontologies of an AmI, and enables the ontological emergence process.**

***Index Terms*— Ontological emergence, reflective middleware, smart environments.**

I. INTRODUCCION

Un conjunto de dispositivos inteligentes o no, que interactúan entre sí, con el objetivo de ofrecer al usuario servicios en apoyo a la realización de sus actividades, constituyen la esencia de un AmI. Estos dispositivos ubicados en el AmI, intercambian datos, información y conocimiento, permanentemente. Para ello, es necesario que puedan tener la capacidad de entenderse, de conocer su contexto, manejar un lenguaje común, entre otras cosas, lo que implica lograr la interoperabilidad semántica [3], [4]. Las ontologías en un AmI constituyen una herramienta ideal para ello, posibilitando la comunicación entre los objetos inteligentes que forman parte del AmI, y la conceptualización del mismo.

Un AmI se puede ver como un entorno donde los diferentes dispositivos que lo componen, y que brindan servicios, usan múltiples fuentes de información, las cuales eventualmente, están en diferentes formatos. La gran cantidad de servicios y dispositivos que se pueden manejar en un momento dado en los AmI, puede generar un gran aumento en el volumen y en la diversidad de los datos, por lo tanto, se necesitan nuevas estrategias que permitan enfrentar de forma eficiente, esta heterogeneidad y dinamismo.

Dr Aguilar has been partially supported by the Prometeo Project of the Ministry of Higher Education, Science, Technology and Innovation of the Republic of Ecuador.

J.L. Aguilar is with CEMISID, Universidad de Los Andes, Mérida, Venezuela. Addtionally, it is Premeteo Researcher at the Escuela Politécnica Nacional, Quito, and the Universidad Técnica Particular de Loja, Ecuador (e-mail: aguilar@ula.ve)

M. Mendonca is with Universidad Centroccidental Lisandro Alvarado, Lara, Venezuela (e-mail: mmendonca@ucla.edu.ve)

N. Perozo is with Universidad Centroccidental Lisandro Alvarado, Lara, Venezuela (e-mail: nperozo@ucla.edu.ve)

Las ontologías que participan en un AmI pueden estar distribuidas (varios servidores) y ser heterogéneas (diversas estructuras de datos, lenguajes y tipos de datos), pero además, deben ser dinámicas, es decir, deben adaptarse a los cambios, necesidades y servicios del AmI.

En los trabajos previos desarrollados [23], [27], hemos podido constatar que los algoritmos tradicionales de minería ontológica por sí solos, no pueden hacer frente a estas necesidades. Para ello se propone, la implementación de un proceso de emergencia ontológica, a través de un middleware reflexivo. El objetivo de la emergencia ontológica, en este caso, es gestionar toda la información y conocimiento que se puede generar o aparecer en un AmI, creando nuevos modelos de conocimiento, que permitan gestionar eficientemente las necesidades del ambiente. Estos nuevos modelos emergentes son usados por los servicios del AmI, para ser más eficientes en el apoyo de las actividades de los usuarios en el ambiente. En particular, lo anterior permite dar respuesta a los requerimientos emergentes en un AmI.

A través de procesos de monitoreo, análisis y enriquecimiento semántico de las diferentes ontologías que se pueden manejar en un AmI, es posible generar modelos de conocimiento emergentes, acordes a su dinámica, y así manejar la escalabilidad presente en un AmI, como también, las necesidades de interoperabilidad semántica.

En este artículo se presenta una arquitectura para el manejo de servicios ontológicos para AmI, que funciona como un KaaS ("Knowledge as a Service", por sus siglas en Inglés) [20], donde el conocimiento que se genera en este caso son ontologías. A diferencia de los OaaS ("Ontology as a Service", por sus siglas en Inglés) propuestos en [17][18][19], nuestra arquitectura, además de integrar, extender o refinar ontologías, es capaz de generar (hacer emerger) ontologías con nuevos conceptos, provenientes del análisis de las necesidades en el AmI y de la información de contexto. La Arquitectura se basa en un marco ontológico compuesto por meta-ontologías, que encaminan el proceso de generación de ontologías (emergencia ontológica), usando conceptos generales y reglas que se deben cumplir, para facilitar la agregación de conceptos provenientes de otras ontologías, de datos capturados del AmI, entre otras cosas.

De esta manera, el objetivo de esta arquitectura es posibilitar procesos de emergencia ontológica, basados en la posibilidad de detectar, almacenar y organizar los modelos ontológicos usados por los múltiples dispositivos del AmI, procesarlos, y generar nuevos modelos ontológicos que se adapten a las necesidades del AmI. Algunas de las utilidades de la arquitectura propuesta son:

- Ofrecer servicios para la auto-gestión de las ontologías del AmI.
- Gestionar de forma inteligente la semántica del AmI, a través de servicios de minería ontológica.
- Modelar los nuevos objetos y comportamientos en el AmI.
- Mantener la consistencia y evolución semántica que el AmI requiera.

En particular, gestionar las necesidades semánticas y las necesidades de nuevas ontologías en un AmI, es una de las tareas fundamentales del middleware. Por ejemplo, el modelado del contexto basado en ontologías es ampliamente usado en la computación pervasiva para obtener el conocimiento del entorno, y en un AmI constituye un poderoso enfoque para lograr razonamiento sobre el contexto [5]. Las ontologías de contexto son por naturaleza dinámicas, y pueden ser generadas por el middleware en base al análisis del comportamiento del AmI.

La arquitectura propuesta del MiR-EO (Middleware Reflexivo para la Emergencia Ontológica) se implementa como un middleware reflexivo, el cual analiza los requerimientos ontológicos en un AmI, y activa los servicios específicos necesarios para que el AmI pueda obrar coherentemente. A través del monitoreo y análisis de lo que ocurre en el AmI, MiR-EO provee servicios ontológicos que realizan los ajustes necesarios del marco ontológico del AmI. Para ello, el middleware maneja un propio marco ontológico interno, conformado por un grupo de meta-ontologías que modelan de forma genérica los elementos (conceptos) que deben contener las ontologías del AmI. Inicialmente, estas meta-ontologías son: la meta-ontología de componentes, la meta-ontología de contexto y la meta-ontología del dominio. Las meta-ontologías de dominio juegan un papel fundamental para la actualización automática de las ontologías de dominio del AmI (en [14] se propone un servicio para esto).

En este artículo se presentan los detalles arquitectónicos, y los componentes, del middleware propuesto. Específicamente, el artículo se estructura de la siguiente manera: la sección II presenta los trabajos relacionados, la sección III las bases teóricas, la sección IV detalla la arquitectura propuesta, la sección V presenta los casos de estudio, y finalmente, la sección VI presenta las conclusiones.

II. Trabajos Relacionados

El manejo de la semántica en un AmI es determinante para el entendimiento del contexto, y para lograr la interoperabilidad semántica entre sus componentes. A continuación, se presentan algunos trabajos donde se considera el diseño de arquitecturas para el manejo semántico en un AmI:

En [6] se propone un middleware para la creación de un entorno inteligente colaborativo en el área de transporte, que se centra básicamente en el manejo inteligente de la información, a través de la ontología del dominio. Ellos proponen un servicio semántico para el descubrimiento dinámico de la información. Además, interopera con servicios implementados en los dispositivos, embebidos en los sistemas de transporte inteligente, para evitar la información redundante y excesiva entre dispositivos, y el ahorro de tiempo. Entre los servicios con los que interactúan el servicio semántico están: vigilancia de tráfico, gestión de semáforos, aviso en paneles, etc. El servicio semántico permite almacenar referencias a objetos y servicios, haciéndolas accesibles de manera semántica, a los dispositivos y aplicaciones del sistema que lo requieran, organizándolas en base a categorías y sub-categorías.

En cuanto a arquitecturas conscientes del contexto para AmI, en [5] se hace referencia a un middleware con capacidad de auto-gestión, apoyado en ontologías de contexto, basadas en tecnologías de la web semántica: OWL (Ontology Web Language, por sus siglas en inglés) y SWRL (Semantic Web Rule Language, por sus siglas en Inglés). Para soportar la auto-gestión, proponen un conjunto de ontologías, estructuradas en

forma de estrella, donde la ontología central es la "Ontología de Dispositivos", que comparte conceptos con la "Ontología de Hardware", "Ontología de Fallas", "Ontología de Calidad" y "Ontología de Servicios", entre otras. La "Ontología de Dispositivos" es la que posee los conceptos generales para la clasificación de los dispositivos del AmI. Cuando hay cambios de estado o llamada a servicios, la información dinámica se introduce en la ontología de contexto auto-gestionada, la cual dispara la ejecución de reglas de auto-gestión como: comprobación de los servicios requeridos por un componente, comprobación de la plataforma de un componente, limitar el número de componentes de un tipo específico de acuerdo a una configuración, entre otras, todo esto, para permitir la adaptación, monitoreo, diagnóstico, y otras tareas propias de la auto-gestión del sistema.

Otra propuesta es [7], donde se propone un middleware para AmI capaz de razonar sobre diversos contextos, modelado en base a ontologías, para lograr una representación semántica del contexto, y así realizar razonamiento en base a ella. Ellos definen en un nivel superior una ontología común, para manejar la información general del contexto, es decir, información sobre: personas, localización, componentes computacionales y actividades. Los detalles de un contexto, se manejan en un conjunto de ontologías en un nivel inferior, para los diferentes dominios, como por ejemplo, el hogar, el vehículo, la oficina, entre otros, las cuales se adaptan o incorporan, dependiendo del ambiente que corresponda.

Una arquitectura orientada a servicio de varias capas para hogares inteligentes, se muestra en [8], donde se modela el contexto en base a: habitantes, localizaciones de los objetos, habitaciones, sensores y actuadores. Además, se monitorea continuamente el ambiente, realizando inferencias sobre el contexto en base a reglas SWRL, para ofrecer los servicios y adaptaciones requeridas por los usuarios. Las reglas son las que determinan qué acciones llevarán a cabo los actuadores del ambiente, y en qué momento, de acuerdo a ciertas condiciones que deben cumplirse, lo que define las capacidades adaptativas al ambiente.

En [9] se presenta una estructura ontológica para AmI, que consiste en tener por un lado los conceptos y sus definiciones, que representan el conocimiento sobre un dominio que no está sujeto a cambios, y por otro lado, las sentencias construidas usando esos conceptos, que representan conocimiento de un problema en específico. Esto permite que la estructura pueda adaptarse al ambiente, y que los agentes en vez de definir en tiempo de diseño como se comunicarán y con qué agentes, sólo especifiquen sus necesidades de información. Posteriormente, aplicando razonamiento ontológico en tiempo de ejecución, pueden inspeccionar la estructura de sus propias ontologías, y de ser necesaria, la de otros agentes, a través de mapeos, para comunicarse.

En cuanto a la aplicación del paradigma de "conocimiento como servicios", en [16] se presenta una propuesta de un arquitectura KaaS para el manejo de datos relacionados a desastres naturales, con el objetivo de ofrecer información eficiente para enfrentar estas situaciones. En la realidad existen una gran cantidad de datos relacionados a desastres disponibles, sobre planes, registros de incidentes, simulaciones y estadísticas. Sin embargo, las soluciones actuales ofrecen muy poca capacidad de integración. El objetivo de aplicar KaaS en este caso, es almacenar la gran cantidad de datos de diversas fuentes, facilitar la generación de conocimiento, y permitir la interoperabilidad. Con respecto a modelos donde se ofrecen ontologías como servicios, se tienen los trabajos [17][18][19] donde se introduce la noción de OaaS ("Ontology as a Service", por sus siglas en inglés). Allí proponen un proceso de adaptación ontológica, como un servicio en la nube. En particular, se proponen servicios de extracción, reemplazo y fusión de ontologías, mediante los cuales múltiples sub-ontologías se extraen desde diferentes ontologías de origen, y luego estas sub-ontologías se fusionan y se extienden, para formar una ontología completa.

En los trabajos revisados se proponen arquitecturas, donde se puede observar el uso de ontologías para la definición conceptual de AmI y para el manejo del contexto. Sin embargo, no se encontraron propuestas que permitan la auto-gestión y adaptación del marco ontológico de AmI, donde las ontologías se puedan actualizar y evolucionar, o puedan emerger, de acuerdo a los cambios del AmI, lo que representa el aporte principal de nuestra propuesta. En [15] se presentó una propuesta inicial de una arquitectura para dar servicios semánticos en AmI basado en 3 ontologías: una Ontología de Componentes, una Ontología de Contexto y una Ontología del Dominio. La arquitectura fue diseñada para ofrecer un conjunto de servicios semánticos, que puedan dar soporte a los procesos que requieren producir y obtener información con contenido y caracterización semántica significativa en el ambiente inteligente.

III. Bases Teóricas

A. Ambientes Inteligentes

Un Ambiente Inteligente (AmI) se define como "aquel que utiliza tecnología computacional para controlar en forma automática su funcionamiento, de manera tal de optimizar el confort del usuario, el consumo de recursos, la seguridad y la eficiencia del trabajo". Se puede decir que son ubicuos o pervasivos, ya que los elementos tecnológicos se insertan en los objetos de uso común, haciendo que la interacción usuario-sistema sea natural y desinhibida [2]. El objetivo de las infraestructuras de computación de un AmI es proporcionar servicios inteligentes a los usuarios, de una manera adaptativa, adecuada a sus necesidades [1]. Para ello, es necesario que los dispositivos sean capaces de entender su contexto, y por su naturaleza heterogénea, necesitan también manejar un lenguaje común, para el logro de la interoperabilidad semántica con los demás componentes del AmI [3] [4].

B. KaaS y OaaS

El paradigma KaaS (Knowledge as a Service, por sus siglas en Inglés), es un modelo de computación en la nube, donde un proveedor de servicios de conocimiento responde a requerimientos de algunos consumidores de conocimiento [20]. El servidor de conocimientos ofrece sus servicios de generación de modelos de conocimiento, que pueden ser costosos e imposibles de obtener por los consumidores de conocimiento. En base a técnicas de minería de datos y de descubrimiento de conocimiento, se ofrecen estos tipos de servicios, basados en el conocimiento extraído de grandes volúmenes de datos, proveniente de múltiples fuentes y con diversos formatos. El

modelo KaaS involucra tres tipos de participantes: proveedores de datos, proveedores de servicios, y consumidores de conocimiento. Los proveedores o propietarios de datos recolectan los datos de sus propios procesos particulares. Los proveedores de servicios ofrecen su servidor de conocimiento, y extraen el conocimiento del conjunto de datos, a través de, por ejemplo, algoritmos de minería de datos. El consumidor de conocimiento consulta un servicio de conocimiento para sus procesos de toma de decisiones. En el caso del modelo OaaS (Ontology as a Service, por sus siglas en Inglés), ellos proveen servicios de adaptación de ontologías en la nube, basadas en procesos de extracción, sustitución y reemplazo de ontologías [25].

C. *Minería Semántica y Minería Ontológica*

La minería semántica se encarga de extraer conocimiento semántico desde diferentes fuentes semánticas, como por ejemplo: páginas web, contenidos estructurados y no estructurados en la web, grafos anotados, ontologías, entre otros [22]. La minería semántica se divide en tres grandes grupos: minería de datos semántica, minería web semántica y minería ontológica. La minería ontológica se refiere a la extracción de patrones de comportamiento y modelos de conocimiento desde un conjunto de ontologías, para lograr un dominio de conocimiento más amplio, enriqueciendo, o construir nuevas ontologías. Actualmente, con el gran crecimiento en las cantidades de ontologías disponibles, es necesaria la minería ontológica, para explorar técnicas que puedan extraer conocimiento global de ellas. Algunas de las técnicas que se han venido desarrollando son de enlazado, mezcla y alineamiento entre ontologías.

D. *Middleware Reflexivo*

Un Middleware Reflexivo es una capa intermedia entre diversas aplicaciones o servicios, donde, a través de la auto-conciencia y auto-referencia, una aplicación puede cambiar su comportamiento según los requerimientos y necesidades del entorno de ejecución. Consta de dos procesos: La introspección, que es la habilidad de observar y razonar sobre su propia ejecución; y la Intersección, que permite la modificación de su propio estado o estructura, como mecanismo de adaptación [12]. Para su implementación se deben considerar dos niveles:

- *Nivel Base:* En el nivel base se encuentran las aplicaciones, donde se ejecutan las funcionalidades propias de las aplicaciones, y los servicios que las conforman. En este nivel es donde se realiza el proceso de intersección, para realizar los ajustes necesarios para modificar el estado de ejecución (estructura), o alterar la interpretación o significado de los datos (comportamiento).
- *Nivel Meta:* En el nivel meta es donde el middleware tiene su capacidad de reflexión, para observar y razonar acerca de los estados de ejecución de las aplicaciones, y determinar cómo adaptar sus estructuras y comportamientos a las necesidades del entorno. La reflexión que se da en este nivel, le confiere al middleware la capacidad de desarrollar sistemas computacionales que pueden ser sensibles a su ambiente.

E. *Computación Autonómica*

La Computación Autonómica es un modelo computacional de auto-gestión inspirado en el sistema nervioso autonómico de los seres humanos [13]. Este paradigma crea sistemas capaces de auto-administrarse, con un alto nivel de auto-gestión transparente para los usuarios. La arquitectura del modelo de computación autonómica se compone de 6 niveles:

- *Recursos Gestionados:* Son los recursos (hardware o software) que son gestionados.
- *Puntos de Enlace:* Conjunto de sensores y actuadores que se incorporan al sistema para gestionar los recursos.
- *Gestor Autonómico:* Implementa el lazo de control inteligente, que automatiza las tareas de autorregulación de la aplicación. Se compone de cuatro fases identificadas como MAPE: Monitoreo (obtiene datos y eventos de los sensores), Análisis (donde se da el diagnóstico), Planificación (se definen las acciones a hacer sobre el proceso) y Ejecución (se envían las órdenes a los componentes a través de los actuadores).
- *Orquestador de Gestores Autonómicos:* Si existen varios gestores autonómicos que necesitan trabajar en conjunto, proporciona el canal de comunicación para la coordinación entre ellos.
- *Manejador Manual:* Permite a los usuarios configurar los gerentes autonómicos, para realizar sus tareas de autogestión, a través de interfaces hombre-máquina.
- *Fuentes de Conocimiento:* Proporciona el acceso a los conocimientos requeridos para la gestión del sistema.

IV. Arquitectura Propuesta

La arquitectura propuesta adopta un enfoque basado en la idea de un KaaS. El conocimiento generado, en este caso, se refiere a "modelos de conocimiento", específicamente a "Ontologías". Esta arquitectura busca la generación de nuevas ontologías, a partir de la recolección y procesamiento de modelos de conocimiento heterogéneos presentes en el AmI, para ponerlas a disposición como un servicio. La idea de basarse en el modelo del KaaS (Ver Fig. 1) es para hacer frente a la variedad de conceptos que manejan los distintos dispositivos que interactúan en el AmI, que comparten el mismo contexto, los cuales serían los proveedores de conocimiento. Este conocimiento, que debe registrarse en el middleware, se integra y procesa en la arquitectura, para ofrecer nuevos modelos conceptuales para el AmI.

En el estudio presentado en [15], se dan unas ideas iniciales de la arquitectura, particularmente, del proceso de emergencia ontológica, donde el contexto es un factor determinante para la generación de ontologías. La arquitectura propuesta (ver Fig. 3) será capaz de hacer emerger ontologías, integrando, enriqueciendo o transformando diversas ontologías, y otros modelos conceptuales, así como datos sensados (captados) y procesados en el AmI, por los dispositivos y servicios que interactúan en él. Para ello, se usaran los servicios ontológicos ofrecidos por nuestra arquitectura, que permiten procesar las diferentes fuentes de información en el AmI, a través de tareas de minería ontológica. Esto permite obtener modelos de

conocimientos adaptables a las necesidades y a la dinámica del AmI, que serán usados por los servicios y dispositivos que en ella subyacen.

A. Arquitectura del MiR-EO

La arquitectura propuesta MiR-EO (Middleware Reflexivo para la Emergencia Ontológica) está inspirada en el paradigma de computación autonómica [13], por lo que se diseña como un middleware reflexivo y autonómico.

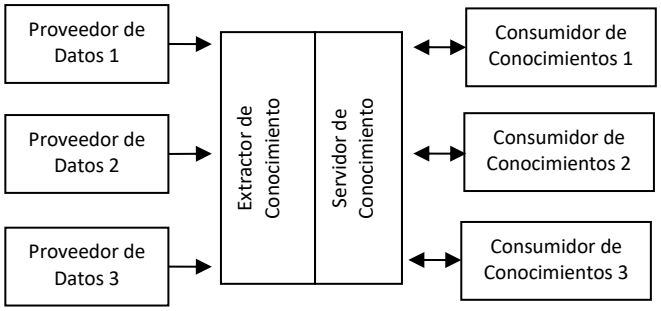

Fig. 1. Paradigma KaaS [21].

La propiedad de reflexión le da la posibilidad de monitorear, tanto los eventos ontológicos (nuevos modelos conceptuales, nuevos conceptos y propiedades), como los servicios que dan soporte al AmI, para adaptarlos y adecuarlos a nuevos requerimientos. Para ello usa su propio marco ontológico, basado en meta-ontologías, que le permite realizar ajustes y auto-gestionar el proceso evolutivo del marco ontológico del AmI. Se está hablando entonces de dos marcos ontológicos: uno que contiene todo el modelo conceptual del AmI, y que se gestiona con MiR-EO, y otro interno del middleware, formado por meta-ontologías, que permiten definir de forma genérica, los conceptos y las reglas que rigen las ontologías del AmI, para apoyar el proceso evolutivo de esas ontologías. En la Fig.2 se puede observar el uso de los dos marcos ontológicos en el gestor autonómico.

En particular, la arquitectura propuesta (ver Fig. 3) consta de los dos niveles propios de los middleware reflexivos:

- *Nivel Base:* En este nivel están todos los componentes del AmI, tanto los recursos (materiales, objetos, usuarios) como los dispositivos (sensores y actuadores). En este nivel se encuentran las aplicaciones y los agentes que ejecutan las funcionalidades del AmI. Tomando como referencia la arquitectura de un AmI propuesta en [28], éste se compone de una Capa Física (CF), una Capa de gestión del SMA (CSMA), una Capa de Gestión de Servicios (CGS), una Capa de gestión del AmI Lógico (CAL) y una capa de gestión del AmI Físico (CAF). Se tiene un modelo conceptual de este nivel, que se define en el marco ontológico de MiR-EO, para la gestión del AmI. A través de los puntos de enlace (sensores y actuadores desplegados en el AmI), es posible gestionar los recursos del AmI (por ejemplo, las ontologías), para realizar los procesos adaptativos que requieran (tales como la inclusión de nuevos conceptos).

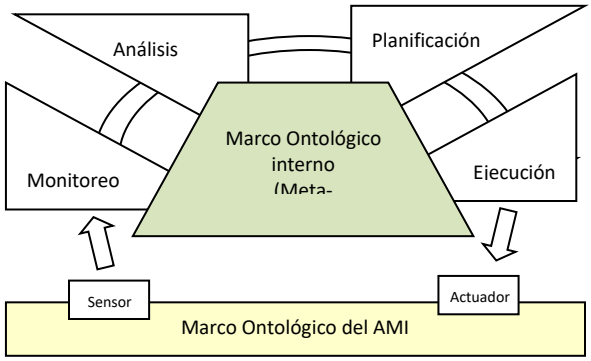

Fig. 2. Marcos ontológicos en el Gestor Autonómico del MiR-EO.

- *Nivel Meta:* En este nivel es donde el middleware tiene su capacidad de reflexión sobre el funcionamiento de sus servicios, y sobre el comportamiento ontológico del AmI, para entre otras cosas, razonar y adaptar su estructura a las necesidades conceptuales del AmI, y de ser necesario, realizar las adaptaciones que requieran las ontologías del AmI. Todo ello es realizado a través de sus servicios ontológicos. Además, es el responsable del proceso de introspección del nivel base. Este nivel se divide en 2 sub-niveles meta:
    - En el Sub-Nivel Meta 1 se encuentran los servicios ontológicos, que de forma colaborativa contribuyen en la generación, actualización y enriquecimiento de las diferentes ontologías que conforman el marco ontológico del AmI, por consiguiente, es donde se ejecutan las funcionalidades de registro semántico, actualización, integración y verificación de ontologías, entre otras. Estos servicios se detallan en la sección D. Además, es donde se da el proceso de emergencia ontológica del AmI. En este nivel es donde se realiza el proceso de monitoreo, a través del registro semántico, para detectar los eventos semánticos en el sistema, y a través del proceso de intersección, realiza las adaptaciones o ajustes necesarios del marco ontológico del AmI.
    - En el Sub-Nivel Meta 2 es donde se estructura el modelo MAPE del Gestor Autonómico del Middleware. A través del monitoreo se detectan necesidades (por ejemplo, la necesidad de generar una nueva meta-ontología). Después se realiza el análisis, para la composición inteligente de los conceptos, propiedades y relaciones que conforman las ontologías del AmI, apoyándose en axiomas y reglas de razonamiento. En este proceso de reflexión juega un papel determinante la meta-ontología que se encuentran en la base de

conocimientos del middleware, que almacenan la información básica sobre la estructura del marco ontológico del AmI.

En esta arquitectura existen un conjunto de componentes distribuidos entre el nivel base y el nivel meta, que trabajan en conjunto para lograr el modelo autonómico del middleware. El manejador autonómico y la base de conocimientos (marco ontológico compuesto por meta-ontologías) trabajan de manera conjunta para realizar la gestión inteligente de sus servicios y de las ontologías del sistema. El marco ontológico, a través de sus meta-ontologías, provee la base de conocimientos necesaria para los procesos de detección de necesidades a nivel semántico, así como para el análisis y planificación de las acciones más adecuadas a realizar.

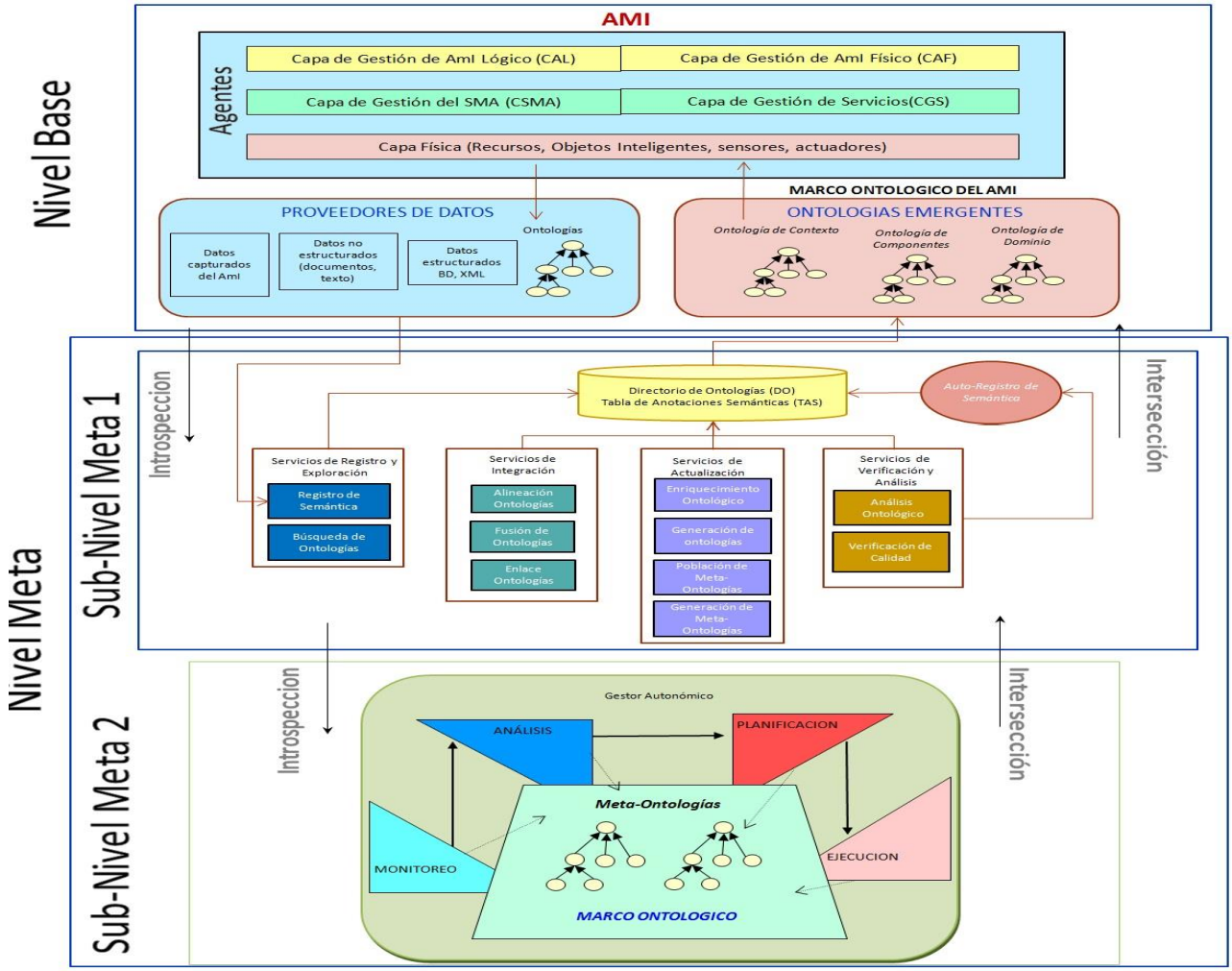

Fig. 3. Middleware Reflexivo para Servicios de Ontologías Emergentes.

B. Estructura del Gestor Autonómico (Modelo MAPE)

El gestor autonómico del modelo propuesto está compuesto por los 4 elementos del modelo MAPE. A continuación se especifican cada uno de los elementos del modelo.

- *Monitoreo:* Se verifican las solicitudes de servicio ontológicos de los dispositivos inteligentes que interactúan en el AmI. El monitoreo se encarga de detectar los eventos semánticos y comportamiento ontológico en tiempo real en el AmI, es decir, qué nuevos modelos conceptuales o qué ontologías están ingresando o saliendo del AmI, y determina la necesidad de registrarlos. Se implementa principalmente, a través del servicio de "Registro Semántico", que genera un directorio semántico de los diferentes componentes que participan en el AmI. Ellos deben registrar sus eventos semánticos, es decir, sus modelos conceptuales (documentos, bases de datos, ontologías). De esta manera, se puede conocer cuál es el estado actual en el ambiente y sus necesidades ontológicas. Estos registros semánticos, así como los procesos de generación e integración de ontologías, se verifican luego, para detectar nuevas necesidades ontológicas, o inconsistencias y redundancias, y así poder gestionar la emergencia ontológica.
- *Análisis:* Se realiza un análisis de lo recogido por el monitor. Se invoca el servicio de "Análisis Ontológico", el cual determina el tipo de necesidad ontológica, y con el servicio de "Verificación de Calidad", se detecta cualquier inconsistencia o redundancia en las ontologías, para aplicar los criterios de calidad correspondientes, y realizar las adaptaciones necesarias. Con estos servicios, se analizan los datos del registro semántico, para detectar cambios en los patrones de comportamiento, así como necesidad de inclusión de nuevos conceptos, nuevas categorías y nuevos modelos conceptuales. Este proceso de análisis puede ser guiado por las meta-ontologías, que ofrecen una clasificación general de categorías de conceptos, y contiene las propiedades y los axiomas que rigen los modelos conceptuales correspondientes a las ontologías que intervienen en el AmI. Realiza también el diagnóstico de cuáles son los posibles nuevos modelos, categorías, comparando el registro semántico con las meta-ontologías.
- *Planificación:* Aquí se evalúan las acciones a seguir para la resolución de las situaciones detectadas en el análisis. En el proceso de planificación se determina cuáles servicios deben ejecutarse para suplir la necesidad de adaptación del marco ontológico del AmI. Determina el tipo de acción a realizar en las ontologías (que tarea de minería ontológica), así como la necesidad de agregar nuevas categorías en las meta-ontologías. Cuando el analizador detecta alguna necesidad se invoca al planificador, para que determine las actividades a realizar. Aquí se determinan las tareas de minería de datos o minería ontológica a realizar.
- *Ejecución:* Aquí se ejecutan los diferentes servicios requeridos que se han planificado, algunos propios del middleware, o de ser necesario se invocan servicios externos. Así, una vez que se ha analizado y se ha planificado en la fase previa las tareas que se realizarán para cubrir las necesidades ontológicas del AmI, viene la fase de ejecución, donde se invocan los servicios necesarios, entre los cuales están los servicios de "Integración de Ontologías" y de "Actualización de Ontologías".

C. Marco Ontológico del Middleware

El marco ontológico del middleware debe ser capaz de modelar las nuevas señales, objetos y comportamientos en el AmI. Aquí se definen la reglas de comportamiento ontológico y las meta-ontologías, que constituyen la base de conocimientos a ser usada por los gestores autonómicos, que permiten la gestión de las ontologías del AmI. Las meta-ontologías definen un conjunto de meta-conceptos o categorías, que pueden ser usados para anotar elementos. Las meta-ontologías son ontologías que contienen conocimiento semántico sobre otras ontologías [10]. Entre las meta-ontologías que se manejan en el marco ontológico se tienen 2 básicas: la meta-ontología de componentes y la meta-ontología de contexto, las cuales son predefinidas. La meta-ontología de dominio en cambio, no puede predefinirse, ya que varía de acuerdo al dominio donde se desenvuelve el AmI, y puede ser definida inicialmente por expertos, e incluso generarse o enriquecerse a partir de las ontologías que participan en el AmI.

1) Meta-Ontología de Componentes

Esta meta-ontología provee la estructura genérica de categorías y relaciones que describen los componentes, los usuarios, las entidades (objetos y dispositivos) que están en el ambiente, que interactúan entré sí, para ofrecer sus servicios o realizar sus actividades y funciones en el AmI. La Fig. 4 muestra las entidades de la meta-ontología de componentes del AmI y sus relaciones. Entre los meta-conceptos que allí se observan se tienen:

- **Recursos:** objetos tangibles (sillas, mesas, computador, etc.) que se encuentran en el AmI. Los recursos pueden ser objetos o contenedores (un contenedor puede contener dentro de sí objetos, por ejemplo, un estante o biblioteca)
- **Espacios:** los recursos están ubicados en diferentes espacios o lugares, que conforman el ambiente (salón, laboratorio, cocina, etc.)
- **Dispositivos:** en los espacios del AmI hay diferentes dispositivos, los cuales eventualmente tienen sus propios sensores y actuadores (cámaras, teléfonos, pizarras inteligentes, etc.).
- **Servicios:** Son las diferentes funcionalidades que ofrecen los dispositivos, que pueden ser usados por los usuarios, recursos u otros dispositivos, en el ambiente.
- **Usuarios:** son los que hacen uso del AmI, de sus recursos, espacios y dispositivos, para la realización de sus actividades, por ejemplo: alumnos, profesores, médicos, entre otros.

2) Meta-Ontología de Contexto

Esta meta-ontología se encarga de caracterizar los conceptos que conforman la información de la ontología de contexto: localización, tiempo, condiciones ambientales, actividades y

perfiles, entre otros. La información del contexto es fundamental, ya que caracteriza a todos los elementos del AmI (definidos en la ontología de componentes), y define los comportamientos y los servicios que deben activarse en el AmI, de acuerdo a la situación y condición actual de sus componentes (usuarios, objetos, entre otros.). La información de contexto es determinante también, para la ubicación de nuevos conceptos en una ontología, ya que por ejemplo, a través de medidas de similitud entre los objetos, en base a sus propiedades y comportamiento, se podrá ubicar en la taxonomía de una ontología de dominio un nuevo concepto. La Fig. 5 muestra los elementos de la meta-ontología de contexto del AmI, y sus relaciones. La meta-ontología de contexto propuesta se diseñó para afrontar la heterogeneidad y permitir la interoperabilidad entre las diferentes ontologías de contexto existentes para un AmI. Esta ontología permite la creación de modelos de conocimientos más precisos y adaptados a la situación cambiante del AmI, contribuyendo así a la unificación de conceptos compartidos entre las diferentes ontologías de contexto. Entre los meta-conceptos que se definen en la meta-ontología de contexto, además de los componentes del ambiente que también forman parte del contexto, se tienen:

- **Estado:** indica el estado en que se encuentra un dispositivo: encendido, apagado, activo, inactivo, en pausa, etc. así como sus propiedades: color, tamaño, temperatura, etc.
- **Perfil:** indica las características de los usuarios o participantes: estado, hábitos, necesidades, preferencias, rol, etc.
- **Condiciones Ambientales:** define características del entorno: iluminación, temperatura, humedad, ruido, presión, etc.
- **Actividad:** caracteriza las actividades que realizan los participantes: su duración, recursos usados, resultados, etc.
- **Tiempo:** determina el momento en que tiene lugar un determinado contexto, indicando día, hora y lapso.
- **Localización:** determina la localización de los usuarios y de los recursos en el ambiente, puede ser, por ejemplo, una localización relativa o absoluta (latitud, longitud).

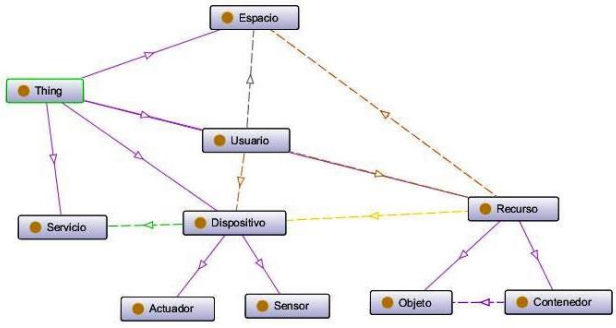

Fig. 4 Meta-Ontología de Componentes

3) Meta-Ontologías de Dominio

En cuanto a las meta-ontologías de dominio de un AmI, estas no pueden ser predefinidas para todos los AmI, ya que dependen de las funcionalidades y objetivos de ellos (por ejemplo, un AmI educativo o médico). Estas meta-ontologías deben ser proporcionadas por fuentes externas (expertos del dominio).

Las Meta-Ontologías, son las que de alguna manera orquestarán todo el conocimiento generado, para mantener la integridad y la coherencia entre los conceptos, ya que ellas guardan dentro de sí la información global y la reglas, de cómo deben estructurarse las ontologías y como deben relacionarse los conceptos entre sí. Las meta-ontologías, pueden servir también para enlazar o localizar ontologías. En base a ellas, se pueden definir o activar, por ejemplo, las ontologías para modelar el contexto de cada uno de los elementos del AmI.

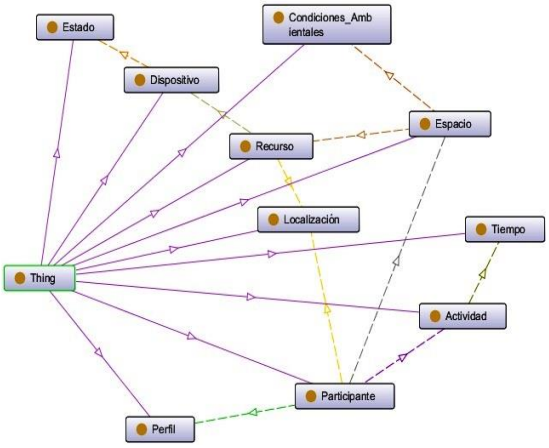

Fig. 5 Meta-Ontología de Contexto

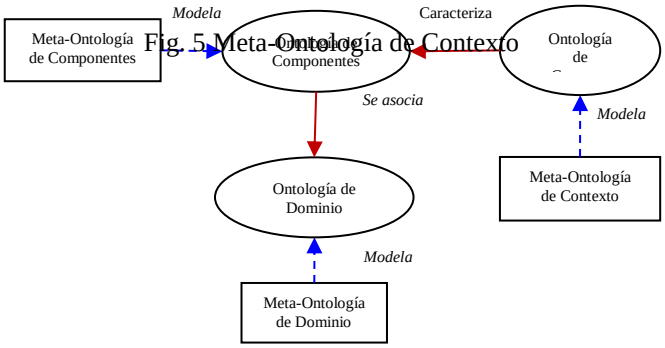

Fig. 6 Arquitectura meta-ontológica

En la Fig. 6 se presenta la arquitectura meta-ontológica que muestra la relación entre las meta-ontologías. Las meta-ontologías definen los conceptos que conforman las ontologías a través de las categorías que definen, es decir, las ontologías son instancias de las meta-ontologías. Por otro lado, la relación entre las ontologías de componentes, de contexto y de dominio es determinante para los servicios del AmI, ya que la ontología de contexto modela el estado en que se encuentran los componentes del ambiente, y a su vez, los componentes en el ambiente están asociados a algún concepto del dominio que caracteriza el AmI. Por ejemplo, el contexto de un usuario en un momento dado viene dado por su perfil, su localización y sus preferencias, lo que determinará los servicios que se pueden activar para ese usuario en ese momento.

4) Reglas de Comportamiento Ontológico

Los axiomas que se definen en las meta-ontologías pueden establecer algunos lineamientos básicos para las ontologías del dominio. Los axiomas son afirmaciones o reglas que permiten restringir la definición de conceptos y sus relaciones, de modo que la definición de los términos del dominio sea más precisa (en este caso, de los conceptos de las ontologías). Los axiomas que se definen determinan: las categorías, las clases disjuntas, las clases complementarias. En la Fig. 7 se muestran un ejemplo de axioma en OWL. Allí se indica en la primera regla que "un sensor es un dispositivo", el cual permite establecer que todos los conceptos que correspondan a "sensores" deben tener las propiedades que posee la clase "dispositivo", definida en la meta-ontología. La segunda afirmación en este caso indica que "un individuo que es miembro de la clase contenedor, no puede ser miembro a la vez de la clase objeto".

En cuanto a los lineamientos generales que determinan las acciones a seguir para lograr la emergencia ontológica, se definen un conjunto de reglas que guiarán el comportamiento ontológico del middleware, entre las cuales están las siguientes:

```
▼<SubClassOf>
    <Class IRI="#Sensor"/>
    <Class IRI="#Dispositivo"/>
  </SubClassOf>
▼<DisjointClasses>
    <Class IRI="#Contenedor"/>
    <Class IRI="#Objeto"/>
  </DisjointClasses>
```

Fig. 7. Axiomas en OWL.

- Los modelos conceptuales que participan en el AmI deben realizar el registro semántico, para lo cual cada concepto debe asociarse a alguna categoría o meta-concepto de las meta-ontologías, determinado por la similitud léxica.
- Si no puede determinarse la similitud léxica, se determina la similitud a través de sus propiedades comunes.
- Si algún concepto no puede ser ubicado o relacionado con ninguna categoría, quiere decir que es un concepto desconocido, por lo tanto, se requiere una posible adaptación de las meta-ontologías con una nueva categoría, lo que representa una evolución del marco conceptual del AmI.
- Si hay más de 2 ontologías del mismo dominio, se puede realizar un proceso de enriquecimiento ontológico (por ejemplo, un proceso de fusión).
- Si hay datos referentes a propiedades de nuevos objetos que participan en el AmI, obtenidos a través de los sensores, se puede determinar sus propiedades y comportamiento, para caracterizar el nuevo concepto y poblar la meta-ontología.
- Si existen más de 2 ontologías alineadas de un mismo dominio, se puede invocar el servicio de generación de meta-ontologías, para obtener nuevos meta-conceptos o categorías que puedan conformar las meta-ontologías.

TABLA I
MACRO-ALGORITMO: SERVICIO DE REGISTRO DE SEMÁNTICA

Entradas: Eventos semánticos: nuevos modelos conceptuales (Bases de Datos, ontologías, documentos xml) .
Procedimiento: 1. Si es una ontología 1.1. Registrar en el Directorio Ontológico (DO) la ontología indicando una descripción y su ubicación. 2. Por cada evento semántico (concepto del modelo conceptual): 2.1. Consultar los meta-conceptos de las meta-ontologías registradas en el middleware (Servicio de Búsqueda de Ontologías) 2.2. Asociar los conceptos del modelo conceptual, con alguna de las categorías (meta-conceptos) presentes en la meta-ontologías. (Servicio de Alineación) 2.3. Registrar en la Tabla de Anotaciones Semánticas (TAS) el evento semántico, en la forma de tripleta: "sujeto","predicado", "objeto. Ej. "Alumno","es un", "Usuario".
Salidas: "Directorio Semántico" (DS) conformado por: la "Tabla de Anotaciones Semánticas" (TAS) y el "Directorio Ontológico"(DO)

TABLA II
EJEMPLO DE DIRECTORIO ONTOLÓGICO (DO)

Ontología	**Ubicación**
"Contenido Programático"	"<http://www.saoye/contenido_prog.owl/>"
"Contexto Auditorio"	"<http://www.saoye/contexto_auditorio.owl/>"

TABLA III
EJEMPLO TABLA DE ANOTACIONES SEMÁNTICAS (TAS)

Sujeto	**Predicado**	**Objeto**	**Fuente**
"Alumno"	"es_un"	"Usuario"	Base de Datos: "Control Estudio"
"Programación"	"es_una"	"Asignatura"	Ontología: Contenido Programático

D. Servicios del Middleware

Los servicios definidos, permiten cumplir los objetivos de: detectar, almacenar y organizar los modelos ontológicos del AmI, dar soporte a los procesos de integración de ontologías, generar nuevos modelos ontológicos (ontologías emergentes), y realizar procesos de verificación de la calidad de las ontologías. El middleware pueden requerir en un momento dado, realizar procesos de auto-gestión, para lo cual, se basarán en las meta-ontologías que componen su base de conocimientos interna, como un modelo genérico, a la hora de componer y actualizar las ontologías requerida por los diferentes procesos del AmI. Los servicios que ofrece el middleware son los siguientes:

1) Servicios de Registro y Búsqueda de Ontología

Estos servicios permitirán conocer y ubicar información semántica en el AmI:

- **Servicio de Registro Semántico:** los dispositivos que participan en el AmI deben, a través de este servicio, registrar información sobre los eventos semánticos, que en este caso se refieren, a los nuevos modelos conceptuales que manejan. De esta manera, se convierten en proveedores de semántica, y a través de este servicio, se suscriben al middleware, que ofrece un entorno de anotación colaborativo, para que los proveedores converjan en un modelo común del conocimiento (Ver Tabla I**)**. Aquí se conforma un Directorio Ontológico (DO) y una Tabla de

TABLA V
MACRO-ALGORITMO: SERVICIO DE ALINEACIÓN DE ONTOLOGÍAS

Entradas: Dos ontologías (AyB).
Procedimiento: 1. Se compara cada concepto de A con los conceptos de B. 2. Se determina si existe similitud morfo-sintáctica. 2.1. Si es así se establece la correspondencia entre los conceptos. 3. Si no existe similitud morfo-sintáctica, se determina si existe similitud léxica. 3.1. Si es así se establece la correspondencia entre los conceptos. 4. Si no existe similitud léxica, se determina si existe similitud semántica. 4.1. Si es así se establece la correspondencia entre los conceptos.
Salidas: Una lista de correspondencias entre conceptos alineados

Anotaciones Semánticas (TAS). Un ejemplo de ellas se pueden ver en las Tabla II y Tabla III.

- **Servicio de Búsqueda de Ontologías:** los dispositivos o agentes que operan en el AmI, pueden requerir localizar ontologías que manejan otros elementos en el ambiente, para localizar conceptos similares o servicios apropiados a sus necesidades. Para ello, el servicio de búsqueda de ontologías le permite explorar el directorio de ontologías del AmI. En este caso, es una búsqueda directa a través del DO (Ver Tabla IV**)**.

2) Servicios de Integración de Ontologías

Dentro del proceso de emergencia ontológica, se requiere llevar a cabo procesos de minería ontológica (alineación, mezcla, enlace), con el objetivo de extraer patrones y conocimiento de las ontologías ya existentes, para luego enriquecer o generar modelos conceptuales integrales y consistentes.

- **Servicio de Alineación de Ontologías:** Consiste en comparar dos o más ontologías, para encontrar correspondencias entre sus entidades (conceptos y relaciones). Busca realizar de forma automática procesos de alineación en base a medidas de similitud (morfo-sintáctica, léxica o semántica). Estas alineaciones permiten realizar procesos de fusión, verificar consistencia entre ontologías, así como encontrar información relevante para procesos de aprendizaje y enriquecimiento ontológico. Igualmente, a través de la alineación se puede realizar el enlazado de distintas ontologías (diferentes nombres con el mismo significado) y detectar similaridad léxica o semántica entre dos descripciones representadas a través de ontologías diferentes (Ver Tabla V**)**.

TABLA VI
MACRO-ALGORITMO: SERVICIO DE FUSIÓN DE ONTOLOGÍAS

Entradas: Dos ontologías (AyB) del mismo dominio, una alineación entre ellas y el tipo de fusión (Debil ó Fuerte).
Procedimiento: 1. Se copia la ontología A en C. 2. Por cada concepto alineado de A con B, se enriquece la ontología C con los nuevos conceptos hermanos y descendientes que no estén en C. 3. Si el tipo de fusión es "Fuerte". 3.1. Se agregan a C todos los conceptos de B que quedaron por fuera.
Salidas: Una nueva ontología (C), producto de la fusión

TABLA VII
MACRO-ALGORITMO: SERVICIO DE ENLACE DE ONTOLOGÍAS

Entradas: Dos ontologías (AyB) del dominios diferentes, una alineación entre ellas.
Procedimiento: 1. Por cada concepto alineado de A con B, se crea el concepto en C, que será el concepto enlace entre los conceptos de A y B alineados enriquece la ontología C con los nuevos conceptos hermanos, ancestros y descendientes que no estén en C. 2. Si no existe alineación entre conceptos, es necesario la intervención de un experto para enlazar conceptos.
Salidas: Una ontología intermedia con conceptos de enlace entre A y B.

- **Servicio de Fusión de Ontologías:** busca la combinación del conocimiento distribuido en varias ontologías, referente a un mismo dominio, para estandarizar el conocimiento o enriquecer el conocimiento ya existente en una ontología. Para ello, se indican cuáles son las ontologías a fusionar, y retornará una ontología con la fusión de ambas. La técnica de fusión también se puede indicar (fusión/mezcla fuerte o mezcla débil) (Ver Tabla VI**)**.
- **Servicio de Enlace de Ontologías:** en el proceso de enlace lo que se busca es encontrar relaciones entre entidades que pertenecen a ontologías de diferentes dominios, para crear una conexión entre ellas sin necesidad de mezclarlas. Esto puede realizarse, identificando las entidades comunes en ellas que sirvan de enlace, y con la creación de una ontología intermedia que permita la navegación entre las ontologías que se están enlazando. En este proceso, si no existe una alineación previa entre conceptos, por lo general puede requerir la ayuda de expertos (Ver Tabla VII**).**

3) Servicios de Actualización de Ontologías

Estos servicios permitirán crear nuevas ontologías, o enriquecer las ya existentes, considerando que la dinámica del AmI demanda un marco ontológico actualizado.

- **Servicio de Población de Meta-Ontologías:** Durante este proceso se realiza un análisis del "Directorio Semántico", para detectar posibles nuevos conceptos que puedan enriquecer las ontologías existentes, o generar nuevas ontologías. Los nuevos conceptos pueden surgir de procesos de comparación y agrupamiento con conceptos de las meta-ontologías, y servirán para poblar las meta-ontologías con esos conceptos (Ver Tabla VIII**)**.
- **Servicio de Generación de Ontologías:** con este servicio es posible la creación de nuevas ontologías, a partir de procesos de aprendizaje ontológico. Aquí se invocan

TABLA IV
MACRO-ALGORITMO: SERVICIO DE BÚSQUEDA DE ONTOLOGÍAS

Entradas: Nombre ó palabras claves para la búsqueda.
Procedimiento: 1. Se realiza la búsqueda en el DO por la descripción de la ontología, con las palabras claves sumistradas. 1.1. Si se consigue, retornar información de ontología. 2. Si no se consigue en el DO, se realiza la búsqueda en la TAS con las palabras claves, comparando con los conceptos (sujetos y objetos) para determinar alguna coincidencia. 2.1. Si se consigue, retornar información de ontología.
Salidas: Información de referencia sobre ubicación de la ontología.

TABLA IX
MACRO-ALGORITMO: SERVICIO DE GENERACIÓN DE ONTOLOGÍAS

Entradas: Fuente de aprendizaje: Texto, BD, xml, ontologías.

Procedimiento:
1. Si la fuente de aprendizaje son datos texto, se invocan procesos de aprendizaje ontológico para el análisis de texto.
2. Si la fuente de aprendizaje son datos :bd, xml, etc, se consulta tabla de anotaciones semánticas y se invoca el servicio de "Población de Meta-Ontologías".
3. Si la fuente de aprendizaje son ontologías se determina si son de dominios diferentes o iguales.
   3.1. Si son de iguales dominios
      3.1.1. Se invoca el servicios de "Alineación de Ontologías"
      3.1.2. Se invoca el servicios de "Fusión de Ontologías"
   3.2. Si son de diferentes dominios
      3.2.1. Se invoca el servicios de "Alineación de Ontologías"

Salidas: Una nueva ontología.

métodos y algoritmos, tanto para la generación de ontologías a partir de fuentes de datos estáticas, como bases de datos, documentos, consultas en lenguaje natural (como el desarrollado en [24]); como para la generación de ontologías emergentes a partir de la información dinámica del ambiente (por ejemplo, a partir de información de contexto [14]). La generación de nuevas ontologías emergentes es activado por el servicio de análisis ontológico, que se realiza sobre el DO y el TAS (Tabla IX**)**.

- **Servicio de Enriquecimiento Ontológico:** en este servicio se busca alimentar las ontologías con nuevos conceptos. Esto se puede lograr a partir de un proceso de alineación y combinación múltiple entre ontologías, para buscar el mayor grado de enriquecimiento. Por ejemplo, usando un algoritmo de inteligencia colectiva como ACO (*Ant Colony Optimization*, por sus siglas en Inglés) [23] (Ver Tabla X**)**.
- **Servicio de Generación de Meta-Ontologías:** si existen más de dos ontologías del mismo dominio alineadas, con

TABLA X
MACRO-ALGORITMO: SERVICIO DE ENRIQUECIMIENTO DE ONTOLOGÍA

Entradas: Un ontología origen y dos o más ontologías alineadas.

Procedimiento:
1. Se aplica el algoritmo de inteligencia colectiva ACO para determinar cuáles son las mejores alineaciones [23].
2. Se selección los conceptos alineados que tienen un mayor medida de similitud
3. Se enriquece la ontología origen con los conceptos hermanos y ancestro de los conceptos alineados seleccionados.

Salidas: Una ontología enriquecida.

TABLA XI
MACRO-ALGORITMO: SERVICIO DE GENERACIÓN DE META- ONTOLOGÍAS

Entradas: Un ontología origen y dos o más ontologías alineadas.

Procedimiento:
1. Se aplica el algoritmo de inteligencia colectiva para determinar propiedades comunes entre conceptos alineados y determinar posibles meta-conceptos[11].
2. Los meta-conceptos se estructuran para conformar la meta-ontología.

Salidas: Una Meta-Ontología.

TABLA XII
MACRO-ALGORITMO: SERVICIO DE ANÁLISIS ONTOLÓGICO

Entradas: Directorio Ontológico (DO) y la Tabla de anotaciones semánticas (TAS).

Procedimiento:
1. Se procesa cada una de las entradas del DO
2. Si hay ontologías sin analizar
   2.1. Se realiza la comparación de otras ontologías, para determinar si son de dominios similares o diferentes (Se invoca el servicio de "Alineación de Ontologías")
   2.2. Si hay 2 ontologías del mismo dominio se sugiere el servicio de "Fusión de ontologías".
   2.3. Si existen más de 2 ontologías del mismo dominio se sugiere el servicio de "Enriquecimiento Ontológico" y el servicio de "Generación de Meta-Ontologías".
3. Se procesa cada una de las entradas del TAS.
4. Se realiza una comparación morfo-sintáctica por cada meta-concepto y los conceptos.
   4.1. Si existe similitud, se sugiere servicio de "Población de Meta-Ontologías"

Salidas: Lista de posibles tareas de minería ontológica.

este servicio se puede realizar un proceso de generación de categorías o meta-conceptos para constituir una meta-ontologías (Ver Tabla XI**)**.

4) Servicios de Análisis y Verificación

Al actualizar el directorio ontológico y la tabla de anotaciones semánticas, es fundamental realizar un análisis para determinar posible conocimiento nuevo, y también verificar la calidad de las ontologías generadas. En este caso, los servicios son:

- **Servicio de Análisis Ontológico:** el servicio de análisis opera sobre el directorio de ontologías para refinar la información semántica recolectada y descubrir nuevo conocimiento (nuevos patrones, conceptos o propiedades) relacionados al dominio, a partir de procesos de minería ontológica. Para detectar nuevo conocimiento, en algunos casos puede ser necesario la aplicación de procesos de aprendizaje semántico (agrupamiento, reconocimiento de patrones, etc.) o alguna técnica de aprendizaje no supervisado, que permita realizar el trabajo de minería semántica para el enriquecimiento de las ontologías. Luego del servicio de análisis se crea una lista de posibles tareas de minería ontológica a aplicar, lo que permite al middleware la generación de ontologías emergente (se orquesta con el servicio de generación de ontologías).
  Este nuevo conocimiento es registrado nuevamente usando el servicio de "Registro Semántico", lo que representa un

TABLA VIII
MACRO-ALGORITMO: SERVICIO DE POBLACIÓN DE META-ONTOLOGÍAS

Entradas: Una Meta-Ontología y la Tabla de anotaciones semánticas.

Procedimiento:
1. Por cada meta-concepto en la meta-ontología, se realiza la búsqueda en el TAS.
2. Se realiza una comparación morfo-sintáctica por cada meta-concepto y los conceptos
   2.1. Si existe similitud, se puede poblar la meta-ontología.

Salidas: Una meta-ontología actualizada con sus instancias

auto-registro de semántica del middleware, para que el nuevo conocimiento pueda ser compartido y reusado por los consumidores de ontologías del middleware (Ver Tabla XII**)**.

- **Servicio de Verificación de Calidad :** para verificar la calidad de una ontología se consideran dos aspectos:
  - *Coherencia:* detectar incoherencias a través de razonadores que permitan determinar si las ontologías son coherentes y no presentan contradicciones.
  - *Redundancia:* para medir el grado de redundancia o solapamiento semántico en una ontología (Ver Tabla XIII).

V. CASO DE ESTUDIO PROPUESTO

A. Definición

En esta sección se presenta el caso de estudio, con el fin de mostrar el funcionamiento de la arquitectura propuesta, y demostrar la capacidad de dar respuesta a situaciones impredecibles en el AmI. Para ello, los escenarios se desarrollarán en un AmI educativo, específicamente, un Salón de Clases Inteligente (SaCI) [26]. El SaCI ha sido modelado usando el paradigma multi-agentes, y ha sido caracterizado usando la arquitectura AmICL [28][29] para desplegarse. AmICL propone dos tipos de agentes, uno para caracterizar los componentes de software, y el otro para definir los componentes de hardware en el ambiente. Básicamente, estos agentes definen las siguientes capas:

- Capa de AmI Físico (CAF): En esta capa se caracterizan los diversos dispositivos presentes en el SaCI (inteligentes o no) como agentes. Ejemplos de posibles agentes en CAF son Pizarra Inteligentes, Robots, etc.
- Capa de AmI Lógico (CAL): En esta capa se caracterizan los diversos componentes de software en SaCI como agentes. Ejemplos de software es el sistema recomendador de contenidos digitales educativos, o el sistema académico.

Las actividades del SaCI son gestionadas por un conjunto de agentes definidos en [29], entre los cuales se tienen:

- *Agente de Visión (AV):* La función principal de este agente es prestar servicios de identificación de los usuarios físicos y virtuales que se conecten (ingresen) o desconecten (salgan) del ambiente.
- *Agente de Dispositivos (AD):* Representa una abstracción lógica de un dispositivo en el ambiente real, lo que permite modelar la interacción con estos objetos.
- *Agente de Gestión (AG):* Modela las características de los diferentes tipos de sistemas de gestión de contenidos, de procesos de aprendizaje, etc. en un ambiente educativo.
- *Agente Tutor (AT):* Este agente es una representación (abstracción) del Profesor o de un entorno de software gestor de procesos de aprendizaje (Sistema Tutorial Inteligente, Ambiente de aprendizaje personal, Entorno Virtual de Aprendizaje) que sirve de guía en el proceso de aprendizaje.

En este caso se propone el uso de la arquitectura, para integrar diferentes modelos conceptuales provenientes de los agetes que participan en el ambiente, para la generación de ontologías emergentes que permitan responder a los requerimientos emergentes del SaCI.

B. Posibles Escenarios

Escenario 1: adaptación de una ontología de contexto con un nuevo participante.

En este primer escenario se quiere mostrar la capacidad de MiR-EO para adaptar una ontología de contexto, al momento de ingreso de usuarios al SaCI. En este caso, al ingresar un tipo de usuario que no está definido en el modelo conceptual para SaCI, el MiR-EO considera el evento como un requerimiento de adaptación o evolución de su modelo conceptual, para la cual invoca los servicios necesarios para generar la emergencia ontológica.

A continuación se describe la secuencia de actividades que realizan los agentes que participan en SaCI en este escenario:

1. El AV supervisa la actividad de un dispositivo de identificación en la entrada del salón, a través de la lectura de código de barra de un AD, el cual autorizará o no el acceso a los usuarios, dependiendo de la actividad que corresponda en ese momento realizar en el SaCI.
2. Este agente, luego de obtener a través del dispositivo de identificación el código del usuario, solicita al AT un servicio de búsqueda de personas pertenecientes a la institución, para poder dar acceso al lugar. Este agente requiere conocer la información de contexto sobre: cuál actividad corresponde a ese ambiente, para ese momento, y cuáles son los usuarios autorizados a entrar.
3. Para ello, el AT requiere la ontología de contexto del SaCI para ese lugar y para ese momento, para lo cual solicita al MiR-EO el servicio de "Búsqueda de Ontologías".
4. La Ontología de Contexto es retornada. En la Fig. 8 se presenta parte esa ontología de contexto, donde se observa que entre las actividades del SaCI se define el concepto "práctica", donde participan "alumno" y "profesor".
5. Al momento de ingresar los usuarios, el AV realiza, a través del servicio de "Registro de Semántica", el registro de conceptos sobre los usuarios, y sus datos para realizar también la población de la ontología con las instancias respectivas (individuos). El registro de semántica tiene el siguiente formato: <Concepto, "IdentificaciónIndividuo">. Por ejemplo AV puede realizar los siguientes registro semánticos: <Alumno, "12322456">, <Alumno,"15678532">, <Profesor,"77996535">.
6. Al momento de ingreso de un tipo de usuario no definido en el modelo conceptual, el evento semántico es detectado (nuevo concepto no definido). El AV determina que está ingresando otro tipo de personal, lo que representa un nuevo concepto en el ambiente inteligente (por ejemplo un "AuxiliarDocente"), al consultar en el modelo ontológico del SaCI y determinar que este tipo de participante no está definido en la ontología de contexto.

TABLA XIII
MACRO-ALGORITMO: SERVICIO DE VERIFICACIÓN DE CALIDAD

Entradas: Una ontología.
Procedimiento: 1. Se realiza el recorrido de la ontología desde el concepto raíz. 2. Se determina si hay conceptos repetidos. 2.1. Si hay conceptos repetidos se registra un tipo de redundancia. 3. Se determina la ruta de recorrido a cada concepto desde la raíz. 3.1. Si se obtiene más de una ruta a un concepto se registra un tipo de redundancia. 4. Se procesa la ontología a través de un razonador para determinar inconsistencia o incoherencia. 5. Si hay inconsistencia o incoherencia se registra
Salidas: Nivel de Redundancia e incoherencia y lista de observaciones.

Se podría simplemente enviar como respuesta, un error durante el "Registro de Semántica", ya que de acuerdo al

modelo definido para ese ambiente, no está definida la participación de este tipo de persona a esa sesión. Sin embargo, como la arquitectura tiene la capacidad de auto-regularse y auto-gestionar sus ontologías, se llevan a cabo las siguientes actividades en el gestor autonómico del nivel meta del middleware:

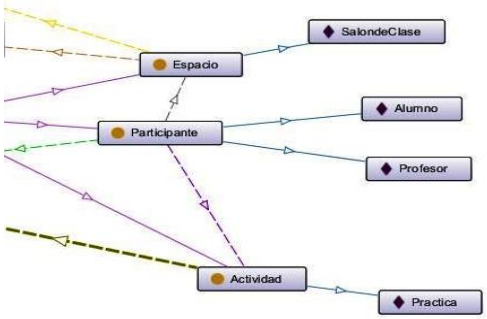

Fig. 8. Ontología de Contexto.

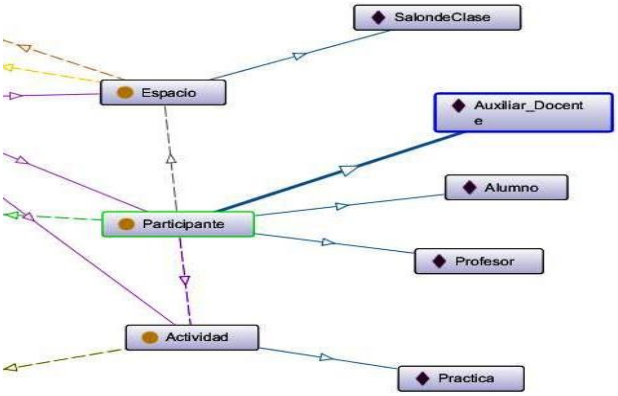

Fig. 9. Ontología de Contexto Actualizada.

- El "Monitor" detecta un evento de error durante el registro de semántica en la Tabla de Anotaciones Semántica, ya que no existe el concepto al que se hace referencia. Esto puede representar un nuevo evento semántico (nuevo concepto), por lo tanto, antes de emitir un mensaje de error, envía al "Analizador" la información para que el determine si ese participante debe ser agregado como un nuevo concepto al modelo ontológico del ambiente.
- El "Analizador" solicita al AT la información sobre el nuevo tipo de participante, para ello este agente debe consultar en la planificación de las sesiones del SaCI, si es posible la participación de este tipo de usuario ("Auxiliar Docente") en esa sesión.
- De ser así, el "Planificador" determina que debe actualizar los conceptos del modelo, ya que "AuxiliarDocente" es un tipo de usuario permitido. Se realiza la consulta en la Meta-Ontología del MiR-EO para determinar la categoría del nuevo concepto, a través de la similitud de sus propiedades. En este caso, se determina que pertenece a la categoría "Participante" de ese AmI.
- El "Ejecutor" solicita el servicio de "Población de Meta-Ontologías", quien realiza entonces la actualización de la Ontología de Contexto del SaCI con el nuevo tipo de participante (Ver Fig. 9).

Escenario 2: adaptación de una ontología de contexto con un nuevo dispositivo

En este segundo escenario se quiere mostrar la incorporación de un dispositivo especial para usuarios con discapacidad en el SaCI, de manera que el contenido que se esté desarrollando se pueda sincronizar de forma adecuada, y estos usuarios puedan participar sin problemas. El proceso se activa cuando MiR-EO detecta el dispositivo especial dentro del SaCI, lo que se considera un evento que requiere servicios de minería ontológica para su incorporación al modelos conceptual del ambiente.

A continuación se describe la secuencia de actividades que realizan los agentes que participan en el SaCI en este escenario:

1. El AD se crea por cada dispositivo que participa en el SaCI.
2. Al momento que un usuario ingresa al salón, y en su dispositivo móvil posee una tecnología especial para el apoyo a su discapacidad visual, el AD a través de algún tipo de sensor, determina el tipo de dispositivo.
3. El SaCI solicita realizar el "Registro semántico" para que pueda formar parte del modelo ontológico del ambiente.
4. El AD tiene que considerar este tipo de dispositivo como parte del contexto, para que en su funcionalidad pueda gestionar el despliegue del contenido de la clase en ese tipo de dispositivo. Para ello, debe usar una estrategia adecuada, para que la clase pueda ser entendido en su totalidad por los participantes con discapacidad visual (por ejemplo, agregar audio a cualquier texto que se presente en la pantalla inteligente, y enviarlo al dispositivo del usuario), activando para ello quizás servicios especiales externos.
5. Para poder adaptar el modelo ontológico se llevarían a cabo las siguientes actividades en el nivel meta del middleware:
   - El "Monitor" detecta en el registro de semántica que no existe el concepto "Dispositivo Especial", en la ontología de contexto.
   - El "Analizador" determinar la necesidad de crear el nuevo concepto "Dispositivo Especial" en la ontología de contexto, y busca información sobre sus propiedades.
   - El "Planificador" determina que debe actualizar en los conceptos del modelo, ya que "Dispositivo Especial" es un tipo de "Dispositivo" de ese AmI.
   - El "Ejecutor" realiza entonces la actualización de la Ontología de Contexto del SaCI con el nuevo tipo de dispositivo.

Escenario3: enriquecimiento ontológico

En este último escenario se quiere mostrar cómo se pueden integrar varias aplicaciones al SaCI y sincronizarse, a través de un proceso de enriquecimiento ontológico. En este caso, el SaCI detecta la solicitud de las aplicaciones y solicita al MiR-EO su incorporación al modelo conceptual.

A continuación, se describe la secuencia de actividades que se realizan en el SaCI en este escenario:

1. Una aplicación para la gestión de aulas virtuales, por ejemplo Moddle, se ha configurado para que se actualice con las sesiones presenciales del SaCI.
2. El AG debe encargarse de coordinar el contenido del aula virtual con el SaCI.
3. Existe una ontología que maneja el SaCI para gestionar la planificación del contenido programático del salón, y otra que maneja el aula virtual para administrar sus cursos. El AG, a través del servicio de "Registro de Semántica", registra las dos ontologías en el MiR-EO.
4. Al registrar las ontologías en el Directorio Ontológico, el MiR-EO en su proceso de gestión autonómica, puede determinar la necesidad de emergencia ontológica de la siguiente manera:
   - El "Monitor" detecta el registro de una nueva ontología en el directorio ontológico, por lo tanto es un evento semántico que debe ser analizado.
   - El "Analizador" determina a través del servicio "Análisis de Ontologías", que en el directorio ontológico hay una ontología que también maneja conceptos de contenidos programáticos. Esta ontología es usada por una aplicación personal para dispositivos móviles, desarrollada por los alumnos para el control individual de sus materias y sus actividades. Se determina entonces que existen estas tres ontologías del mismo dominio, que pueden ser fusionadas. En la Fig.10 se observan las 3 ontologías.
   - El "Planificador" determina que se debe solicitar el servicio de "Alineación de ontologías", y luego el servicio de "Enriquecimiento de Ontologías" entre la ontología del SaCI y los otras dos ontologías, para obtener una nueva ontología enriquecida. Igualmente, es posible invocar el servicio de "Generación de Meta-Ontologías", para determinar si existen nuevas categorías o meta-conceptos.
   - El "Ejecutor" solicita el servicio de alineación y enriquecimiento de ontologías, y luego de generación de meta-ontologías. La nueva ontología enriquecida (Ver Fig.11) es registrada en el directorio ontológico, para ponerla a disposición de los servicios del SaCI.

VI. CONCLUSIONES

En este artículo se propone una arquitectura para la emergencia ontológica en AmI, cuyo objetivo es ofrecer un conjunto de servicios que permiten generar ontologías de acuerdo a las necesidades del AmI. Se implementa como un middleware reflexivo, el cual analiza los requerimientos ontológicos, y activa los servicios de minería ontológica necesarios para que se dé el proceso de emergencia ontológica. Este middleware maneja su propio marco ontológico interno, conformado por un grupo de meta-ontologías que modelan de forma genérica los elementos (conceptos) que deben contener las ontologías del AmI. A través del proceso de reflexión sobre el funcionamiento del AmI, se puede interceder para realizar las adaptaciones y la auto-regulación que requiera el marco ontológico y así mantener la consistencia y evolución semántica que el AmI requiera.

La arquitectura propuesta fue validada a nivel de diseño, a través de 3 casos de estudio, los cuales permiten verificar las funcionalidades definidas en el middleware. Actualmente se viene trabajando en casos reales experimentales, y en la definición de métricas que permitan evaluar el comportamiento del middleware, que serán reportados en próximos trabajos

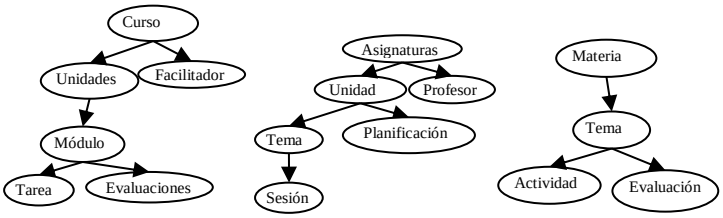

Fig. 10. Diferentes Ontologías del dominio educativo (a) Ontología de un Aula Virtual b) Ontología SaCI c) Ontología Disp. Personal)

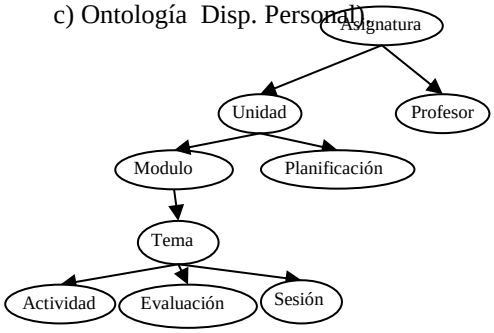

Fig. 11. Ontología enriquecida del SaCI.

La arquitectura propuesta, muestra una capacidad de reacción y adaptación a situaciones específicas donde se presenten necesidades particulares, que pueden ser resueltas a través de procesos de emergencia ontológica en los diferentes servicios que se ofrecen. El aporte más resaltante del artículo, es la propuesta del middleware reflexivo para la gestión de ontologías en un ambiente inteligente, el cual permite la emergencia de ontologías según las dinámicas que se van presentando en dicho ambiente.

REFERENCIAS

[1] J. Aguilar, "Introducción a los Sistemas Emergentes", *1ra Edición, Talleres Gráficos*, Universidad de Los Andes, 2014.

[2] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, and K. De Bosschere, "Towards an extensible context ontology for ambient intelligence", *Ambient intelligence*, pp. 148-159, 2004.

[3] L. González and J. Echeverri, "Modelado Conceptual de Usuarios en Ambientes Ubicuos Mediante Agentes Y Ontologías", *Revista EIA*, pp. 115-126, nro. 16, 2011.

[4] J. Kiljander, A. Ylisaukko-oja, J. Takalo-Mattila, M. Eteläperä, and J. P. Soininen, "Enabling semantic technology empowered smart spaces", *Journal of Computer Networks and Communications*, pp. 1-14, 2012.

[5] T. G. Stavropoulos, D. Vrakas, D. Vlachava, and N. Bassiliades, "BOnSAI: a smart building ontology for ambient intelligence", *Proc. of the 2nd International Conference on Web Intelligence, Mining and Semantics*, pp. 30-41, 2012.

[6] K. Hansen, W. Zang, J. Fernandes and M. Ingstrup, "Semantic web ontologies for ambient intelligence", *Proc. of the 1st International Research Workshop on The Internet of Things and Services*, pp. 1-6, 2008.

[7] D. Gregor, "Desarrollo de un Servicio Middleware de Ontologías Cooperativas aplicado a Sistemas Embebidos de Transportes Inteligentes", *Tesis Doctoral*, Universidad de Sevilla, Sevilla,2013.

[8] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang "An ontology-based context model in intelligent environments", *Proc. of communication networks and distributed systems modeling and simulation conference*, vol. 2004, pp. 270-275, 2004.

[9] V. Ricquebourg, D. Durand, D. Menga, B. Marine, L. Delahoche, C. Loge, and A. M. Jolly-Desodt "Context inferring in the Smart Home: An SWRL approach", *Advanced Information Networking and Applications*, vol. 2, pp. 290-295, 2007.

[10] J. Diggelen., R. Beun, R. van Eijk, and P. Werkhoven "Efficient semantic information exchange for ambient intelligence", *The Computer Journal*, pp. 1138-1151, 2010.

[11] Guizzardi, G. "On ontology, ontologies, conceptualizations, modeling languages, and (meta) models", *Frontiers in artificial intelligence and applications*, vol. 155, pp. 18-28, 2007.

[12] P. Maes "Concepts and Experiments in Computational Reflection", *Proc. of ACM Conference on Object-Oriented Programming*, vol. 22, pp. 147-155, 1987.

[13] IBM Autonomic Computing Architecture Team, "An architectural blueprint for autonomic computing", Technical report, IBM Corporation, Hawthorne, NY, Fourth Edition, Junio 2006.

[14] M. Mendonça, J. Aguilar, and N. Perozo, "Una Ontología Emergente para Ambientes Inteligentes basada en el Algoritmo de Optimización por Colonias de Hormigas", *Proc. of The Latin American Computing Conference*, pp. 596-606, 2014.

[15] M. Mendonça, J. Aguilar, and N. Perozo, "Middleware Reflexivo Semántico para Ambientes Inteligentes", *Conferencia Nacional de Computación, Informática y Sistemas*, pp. 24-32, 2014.

[16] K. Grolinger, M. Capretz, E. Mezghani, and E. Exposito, "Knowledge as a service framework for disaster data management", *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd International Workshop*, pp. 313-318, 2013 .

[17] K. Grolinger, E. Mezghani,M. Capretz and E. Exposito, "Collaborative knowledge as a service applied to the disaster management domain", *International Journal of Cloud Computing*, vol. 4, no 1, pp. 5-27, 2015.

[18] A. Flahive, D. Taniar, and W. Rahayu, "Ontology as a Service (OaaS): a case for sub-ontology merging on the cloud", *The Journal of Supercomputing*, vol. 65, nro. 1, pp.185-216, 2013.

[19] A. Flahive, D. Taniar, and W. Rahayu, "Ontology as a Service (OaaS): extracting and replacing sub-ontologies on the cloud", *Cluster computing*, vol.16, nro. 4, pp. 947-960, 2013.

[20] A. Flahive, D. Taniar, and W. Rahayu, "Ontology as a Service (OaaS): extending sub-ontologies on the cloud", *Concurrency and Computation: Practice and Experience*, vol. 27, nro. 8, pp. 2028-2040, 2015.

[21] S. Xu, and W. Zhang, "Knowledge as a service and knowledge breaching", In *Services Computing, IEEE International Conference on*, vol. 1, pp. 87-94, 2015.

[22] Q. Quboa, and M. Saraee, "A State-of-the-Art Survey on Semantic Web Mining", *Intelligent Information Management*, vol 5, pp. 10-17, 2013.

[23] M. Mendonça, J. Aguilar, and N. Perozo, "An approach for Multiple Combination of Ontologies based on the Ants Colony Optimization Algorithm", *Proc. of Asia-Pacific Conference on Computer Aided System Engineering*, pp. 140-145, 2015.

[24] T. Rodríguez, and J. Aguilar, J. "Ontological learning for a dynamic semantics ontological framework", *Dyna*, vol, 81, nro.187, pp. 56-63, 2014.

[25] S. Sharma (2015). Evolution of as-a-Service Era in Cloud. Iowa State University, USA [Online]. Disponible: https://arxiv.org/ftp/arxiv/papers/1507/1507.00939.pdf .

[26] P. Valdiviezo, J. Cordero, J. Aguilar, and M. Sánchez, "Conceptual Design of a Smart Classroom Base on Multiagent System", *International Conference on Artifitial Intelligence ICAI'15*, pp. 473-477, 2015.

[27] C. Rangel, J. Aguilar, M. Cerrada, J. Altamiranda, "An Approach for the Emerging Ontology Alignment based on the Bees Colonies", *Intl. Conf. Artificial Intelligence (ICAI'15)*, pp. 536-541, Julio 2015

[28] M. Sánchez, J. Aguilar, J. Cordero, and P. Valdiviezo, "A Smart Learning Environment based on Cloud Learning", *International Journal of Advanced Information Science and Technology (IJAIST*), vol. 39, nro.39, pp. 39-52, 2015.

[29] M. Sánchez, J. Aguilar, J. Cordero, and P.Valdiviezo, "Basic features of a Reflective Middleware for Intelligent Learning Environment in the Cloud", *Asia-Pacific Conference on Computer Aided System Engineering (APCASE'15)*, pp. 1-6, 2015.

Maribel Mendonça es Ingeniero en Informática graduada en 1998 en la Universidad Centroccidental Lisandro Alvarado, Barquisimeto, Venezuela. M.Sc. en Ciencias de la Computación en 2004 de la Universidad Centroccidental Lisandro Alvarado, Venezuela. Actualmente es Profesora Agregada en el Departamento de Sistemas de la Universidad Centroccidental Lisandro Alvarado, Venezuela. Su área de investigación incluye Inteligencia Artificial y Aprendizaje Ontológico.

José Aguilar es Ingeniero de Sistemas graduado en 1987 en la Universidad de los Andes, Mérida, Venezuela. M.Sc. en Ciencias de la Computación en 1991 de la Universidad Paul Sabatier-Toulouse-Francia. Ph.D en Ciencias de la Computación en 1995 de la Universidad de René Descartes-Paris-France. Completó estudios de post-doctorado en la Universidad de Houston. Es investigador en el Centro de Microcomputación y Sistemas Distribuidos (CEMISID) en la Universidad de los Andes. Miembro de la Academia de Ciencias de Mérida y del Comité Técnico Internacional del IEEE-CIS en Redes Neuronales Artificiales.

Niriaska Perozo es Ingeniero en Informática graduada en 1997 en la Universidad Centroccidental Lisandro Alvarado, Barquisimeto, Venezuela. M.Sc. en Ciencias de la Computación en 2004 de la Universidad de los Andes, Mérida, Venezuela. Doctora en Ciencias Aplicadas (Universidad de los Andes, Mérida, Venezuela) y Neurociencia, Cognición y Comportamiento Colectivo (Universidad Paul Sabatier, Toulouse, Francia) en 2011. Actualmente es Profesora e Investigadora en Inteligencia Artificial en la Universidad Centroccidental Lisandro Alvarado, Venezuela.

Setting a generalized functional linear model (GFLM) for the classification of different types of cancer

Miguel Flores, Guido Saltos and Sergio Castillo-Páez

Abstract **— This work aims to classify the DNA sequences of healthy and malignant cancer respectively. For this, supervised and unsupervised classification methods from a functional context are used; i.e. each strand of DNA is an observation. The observations are discretized, for that reason different ways to represent these observations with functions are evaluated. In addition, an exploratory study is done: estimating the mean and variance of each functional type of cancer. For the unsupervised classification method, hierarchical clustering with different measures of functional distance is used. On the other hand, for the supervised classification method, a functional generalized linear model is used. For this model the first and second derivatives are used which are included as discriminating variables. It has been verified that one of the advantages of working in the functional context is to obtain a model to correctly classify cancers by 100%. For the implementation of the methods it has been used the fda.usc R package that includes all the techniques of functional data analysis used in this work. In addition, some that have been developed in recent decades. For more details of these techniques can be consulted Ramsay, J. O. and Silverman (2005) and Ferraty et al. (2006).**

Index Terms**— Depth of functional data, DNA, functional data analysis, functional distances, statistical classification**

I. Introduction

The DNA Microarray chips and high-density oligonucleotide are widely used in modern biomedical research and can serve as a guide for the diagnosis and treatment of some diseases.

One of the most interesting and current applications is the characterization and classification of different types of cancer Singh D. et al. (2002). Microarray data show expression levels of many genes with respect to a number of observations (samples) and therefore can be considered as functional data or data with high dimension.

To this effect, it is very common to use multivariate methods to classify or create groups, for example according to Romualdi et al., (2003); Wessels et al., (2005); Tárraga et al. (2008) the best methods are: the K nearest neighbor method (KNN) and Diagonal Linear Discriminant Analysis (DLDA). Also, in the work of Dudoit et al. (2002) you can see a comparison of discrimination methods for the classification of tumors using gene expression data.

However, these methods of classical statistics do not perform well when the dimension of the data is very high relative to the size of the sample. (López-Pintado et al., 2010)

In this paper, a new approach is proposed for the classification of different types of cancer using models of Functional Data Analysis (FDA). This recent field of statistics allows processing data with high dimension and take advantage of their functional character.

Specifically, it is used a generalized functional linear model fit to classify the levels of expression of a set of genes in a type of tumor that affects a group of individuals.

To illustrate procedures of Functional Analysis of data is used, the database "prostate" belonging to the package " depthTools " R, which contains a random sample of 25 non-tumor samples (healthy) and 25 tumor samples (malignant), in which have been measured the expression levels of 100 genes. For more details on the data, you can consult Singh D. et al. (2002).

A glossary follows explained in Table I.

TABLE I
TERMS GLOSSARY

Term	Definition
AIC	Akaike Information Criteria
DNA	Deoxyribonucleic Acid
DLDA	Diagonal Linear Discriminant Analysis
FDA	Functional Data Analysis
FPLS	Functional Partial Least Squared - Principal Component
FM depth	Fraiman and Muniz depth
GCV	Generalized Cross-Validation
GFLM	Generalized Functional Linear Model
KNN	K nearest neighbors estimator
LLR	Local Linear Smoothing
NW	Nadaraya Watson Kernel Estimator
PL	Partial Least - Principal Component
RP depth	Random Projection depth

Finally, to implement the FDA procedures, the R statistical software is used, because the R package fda.usc has applicable routines for functional data. This package carries out exploratory and descriptive analysis of functional data, analyzing its most important features such as depth

Miguel Flores, is a professor at the Departamento de Matemática, Escuela Politécnica Nacional, 17012759 Ecuador (e-mail: miguel.flores@epn.edu.ec)
Guido Saltos, is a professor at the Universidad de las Américas, Quito, Pichincha, Ecuador (e-mail: guido.saltos@udla.edu.ec)
Sergio Castillo-Páez, is a professor at the Universidad de las Fuerzas Armadas del Ecuador ESPE, Sangolqui, Pichincha, Ecuador (e-mail: sacastillo@espe.edu.ec)

measurements or functional outliers detection, among others. Besides, fda.usc includes the functions implemented by Ferraty et al. (2006).

II. Functional Definition And Representation

$\mathcal{X}$ is defined as functional variable of interest, level of expression of genes taking values in a normed space (or semi - normed) $\mathcal{F}$, and the set $\{x_1, x_2,...,x_n\}$ is considered the functional data to be analyzed which come from n functional variables $X_1, X_2,...,X_n$ identically distributed as $\mathcal{X}$. Functional data are discretized in a set of points $\{t_j\}_{j=1}^{d}$ not necessarily equidistant (as here).

Therefore, it has d (genes) assessments for each of the n (observations) functional variables, that is, with a matrix of 50 rows representing discretized curves and 100 columns representing points to evaluate. The first 25 rows correspond to levels of expression of normal tumors and the following 25 rows to malignant tumors.

In Figure 1, you can see in black the different levels of the genes for normal tumors and red for malignant. At first glance this figure does not distinguish differences between tumor types.

To appreciate a greater difference on the relationship of genes and their expression level, for each tumor type a panel of six graphs is presented in Figure 2, in each row there are three graphs corresponding to normal tumors, first row, and malignant tumors, second row. The graphs in each row corresponds respectively to functional data (first), first derivative (second), and second derivative (third).

The representation made in Figure 1 for the functional data implicitly assumes a space L_2 which does not allow adequate discrimination between tumor types; you can see that by studying the behavior of the level of gene expression in other spaces (see Figure 2) can have a better discrimination. Specifically, the functional space of the second derivative of the functional data provides greater features (depth and variability) to discriminate between the two tumors.

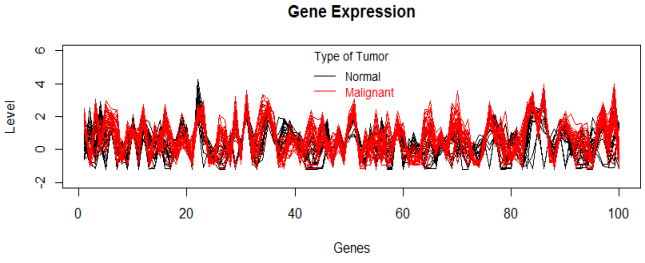

Figure 1: Graph of functional data represented by curves of black color for normal and red for malignant tumors.

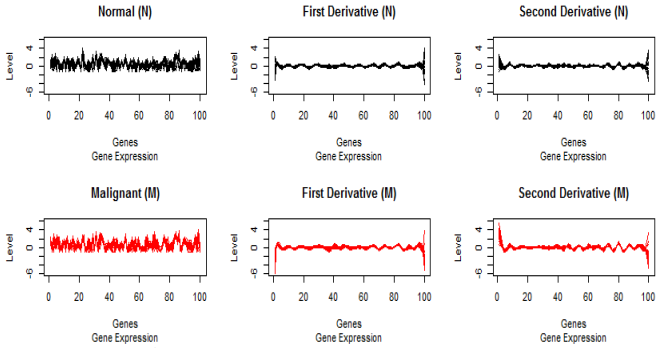

Figure 2: Panel of six graphs, in each row there are three graphs corresponding to the functional data, first derivative and second derivative; in the first row for normal tumors and in the second row for malignant tumors.

The representations in bases made for the first curve of the sample's functional data are shown in Figure 3: B-Splines (5, 20), Kernel Smoothing (KNN, LLR, NW) and Principal Components (PL and FPLS). These representations allow you to work the problem in finite dimension.

For the selection of a base, a setting parameter must be calibrated that allows a better representation; for this selection has been considered as criterion the Generalized Cross-validation (GCV) method. For more information about base types, methods and validation criteria, see Febrero-Bande, M. and Oviedo de la Fuente, M. (2012).

For the classification of tumors we work with representations in base; but for calculating distances and exploratory analysis of functional data we do not work with representation in base. The fda.usc R package is used to perform calculations using the corresponding numerical approximations.

As can be seen in Figure 3, depending on the method and the adjustment parameter representations in base, they are different. In the case of a representation by principal components we can see that there is not much difference between the PL and PLS method.

In Table II the following indicators are shown: the percentage of variance explained for each component; the correlation between the level of gene expression; and the type of tumor. These indicators are calculated for the original data, its first derivative and second derivative.

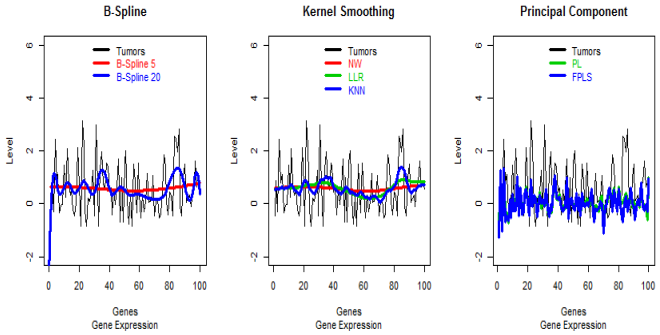

Figure 3: The base representations made to the first observation of the functional data with bases: B -Splines (5.20), Kernel Smoothing (KNN, LLR, NW) and Principal Components (PL and FPLS).

TABLE II
PERCENTAGE OF EXPLAINED VARIANCE FOR EACH COMPONENT AND THE CORRELATION BETWEEN THE EXPRESSION LEVEL OF GENES AND TUMOR TYPE FOR THE ORIGINAL DATA, ITS FIRST DERIVATIVE AND SECOND DERIVATIVE

Data	Method	Indices	PC1	PC2	PC3
Original	PL	% explained variance	75.21	17.8	6.98
		Tumor type correlation	67.8	-37.6	-11.3
	FPLS	% explained variance	79.75	14.78	5.47
		Tumor type correlation	75.1	41.5	30.9
First Derivative	PL	% explained variance	71.85	23.76	4.39
		Tumor type correlation	68.8	-35.6	41.6
	FPLS	% explained variance	75.3	18.64	6.06
		Tumor type correlation	74.1	41.4	60.4
Second Derivative	PL	% explained variance	68.8	22.22	5.96
		Tumor type correlation	66.8	-41.7	38.4
	FPLS	% explained variance	74.27	19.85	5.88
		Tumor type correlation	73.7	42.6	52.2

About 70 % of the total variability of the data is explained by the first component, regardless of the method of principal components to be used; the variability explained by the second component increases to about 20 % when working in the spaces of the functions of the first and second derivatives.

In general, we can say that the first two components explain about 90 % of the variability; the first component has a strong positive ratio of about 70 % in all methods; and the second component has a negative ratio using the method PL and a positive one using the PLS method.

III. Distance Between Functional Data

In this section it has been applied a metric for the L_2 space and 4 semi - metrics for other semi - normed spaces, in order to calculate the distance between the functional data (for more information on the definition of each measure, see Febrero-Bande, M. and Oviedo de la Fuente, M. (2012). For calculating these measures, have been implemented the following functions developed in the fda.usc package:

1) metric.lp (for functional data represented in a L_p space, with p = 2).
2) semimetric.deriv (for functional data in the space of functions of the first and second derivative).
3) metric.pl (based on the method of principal components (PL), it calculates a PL semi- metric between functional data).
4) metric.mplsr (based on the principal component method (PLS), it calculates a FPLS semi- metric between functional data).

TABLE III
PERCENTAGE OF CORRECT CLASSIFICATION OF TUMORS FOR EACH METHOD.

Function	Space	% success
metric.deriv	First derivative	40
metric.deriv	Second derivative	78
metric.pca	Principal Component PL	84
metric.mplsr	Principal Component FLPS	98
metric.lp	L^2	40

The result of each of these functions (metric and semi- metric) is a matrix of dimension 50 x 50 containing the distances between all curves (functional data).

You can use this information as a classification rule, since it is expected that the closest curves belong to the same tumor. In Figure 4 the dendrogram for the semi - metric Principal Component FPLS is shown.

The results presented in Table III, are the percentages of correct classification and correspond to distance functions (the metric and semi- metric). The highest values are those calculated by the metric.lp and metric.deriv functions. With the semi - metric calculated by the metric.mplsr function it was achieved a 98 % of success; it should be mentioned that only came to classify erroneously one case (curve 40: a malignant cancer classified as normal). Cuevas et al., (2001) use an approach based on density estimation for doing a Cluster Analysis.

Clearly, with these results is more advisable to work in semi - normed spaces to identify differences between the expressions of genes according to cancer types for better classification.

IV. Exploratory Analysis Of Functional Data

For this section, the exploratory data analysis has been divided in two parts:

1) Variability and central tendency estimation
2) Outliers detection

Estimates of central tendency and variability for each type of cancer are done using robust methods, therefore there is not a great influence by outliers for estimates.

However, it was decided to conduct a study of outliers detection to illustrate the methodology to be used in the case of not having these robust methods.

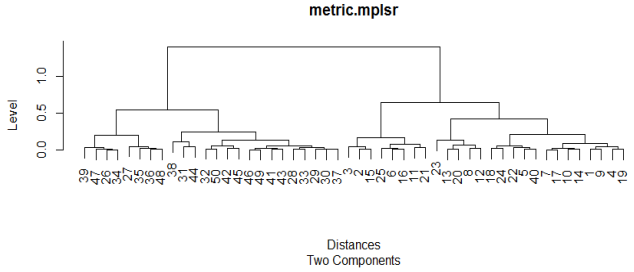

Figure 4: Dendrogram for the semi - metric Principal Component FPLS (98% correct classification)

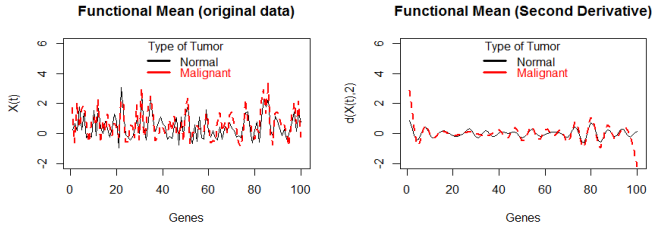

Figure 5: Functional Mean for original data and second derivative for each type of tumor.

A. *Variability and central tendency estimation*

The descriptive exploratory analysis consists in: calculating the mean, and functional variance of the expression levels of genes and its second derivative. This study is performed for each type of tumor to differentiate central tendency and variability of functional data.

In Figure 5, we can appreciate the functional mean for the original data and its second derivative. In each graph the functional mean is distinguished for each type of cancer.

As shown in the graph on the left, the difference between the curves of the functional means for the original data is not very noticeable; on the other hand, in the right picture for functional mean of the second derivative you can see a greater difference.

Overall the two graphs give us an idea that the expression levels of genes tend to values between -0.5 and 1.5, approximately; whereas, the second derivative between -0.5 and 0.5. In addition, we can see that there is greater variability in the trend of the original data than in the second derivative.

In Figure 6, it is shown a graphical representation of the confidence ball representing the estimation limits, where the functional mean oscillates for each type of functional data in each space. It has been applied the smoothing bootstrapped method using the "fda.booststrap "function included in the fda. usc package.

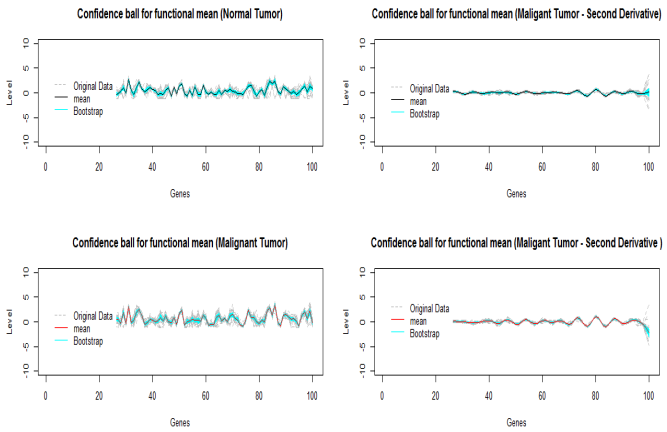

Figure 6: Confidence balls for Functional Mean for original data and second derivative for each type of tumor.

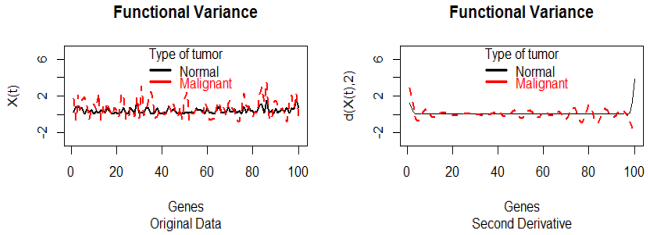

Figure 7: Functional variance for the original data and second derivative for each type of tumor

In Figure 6, light blue curves are the representation of confidence balls at a level of 95% generated by the bootstrap method for functional mean (black color curve). For graphics on the left, gray curves correspond to the original data and for graphs on the right, they are the second derivative's.

In Figure 7 the curves of variance for each type of cancer are shown in the spaces of the original data (graph on the left) and the second derivative's (right graph). Here you can see a marked difference between the variance of normal and malignant tumors. In malignant tumors a greater range of variation is observed that in normal ones; this same behavior is similar in the two spaces of functional data.

B. *Outliers Detection*

Subsequently, a study on the presence of outliers is done because they could affect the estimation and performance (classification) of the model. The depth is a measure whose concept has emerged in the literature of robustness, measures how deep (or central) is a benchmark for a population (or sample). Therefore, those points having large depth values, will be closer to the behavior of the central data; and if they have less deep values, they will be potential candidates for outliers. For more information about the definition of a function of depth see Zuo Y. and Serfling R. (2000).

In univariate data, the median would be the deepest point of the set of points. For this study, we have applied the following depth measures which are included in the package fda.usc: Mode (mode depth); Median defined by Fraiman (Fraiman and Muniz, 2001) (FM depth); and Random Projections (RP depth).

Having studied the central tendency and variability of the data we continue with the detection of outliers in the sample. We start with an analysis with all the original data by calculating three measures of depth (shortened by 10 %) and the difference of each with respect to the median of functional data is observed (see Figure 8); subsequently, a scatter plot is made between the different depth measurements to see if there are outliers (the points with smaller depth values and that are not aligned to the general behavior of the points are considered outliers).

The analysis for the detection of outliers is accounted considering all the sample data; but this analysis is applicable for each subsample defined by the type of tumor. Table IV summarizes the curves (or outliers) considering the total sample and the subsample for each type of cancer.

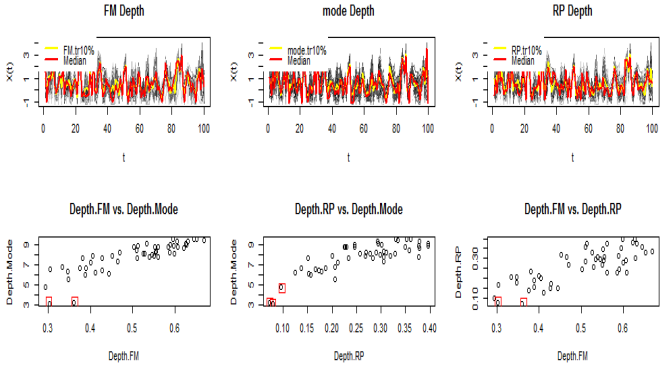

Figure 8: A panel of six graphs, the first row has the representation of depth measurements contrasted with the median of the total sample. The second row shows the scatter plots between all the depth measurements.

In the code attached to this work can be found the procedure developed for the calculation of depth measurements as their graphic representation for the entire sample and for each subsample.

In Figure 8, a panel of six charts that are distributed as follows is presented: in the first row is the representation of depth measurements contrasted with the median of the total sample; and in the second row, we have the scatter plots between all the depth measurements; it should be noted that have been marked with a red box the outliers for each graph and the order considered for reading the graphs is from left to right.

Contrasting depth measurements (second row of the graphics panel of Figure 8), clearly can be observed outliers in all three cases. In the first graph two points are identified as atypical functional data representing curves 2 and 21 belonging to the normal tumor sample; the same curves also are identified as atypical curves by observing the third graph; while in the middle graph three atypical points are observed, curves 2, 21 and 3.

To confirm this visual analysis, an analytical rule is applied which considers as atypical functional data the curves whose depth values are less than a quantile defined based on all calculated values of depth of each sample's data (curves).

In the case of the mode depth measurements to a 1% quantile, it could be identified as atypical data curves 2 and 21; this also happens with the depth measurement of random projections, i.e., the curves 2 and 21 are identified as atypical again with 1% quantile. Whereas, for the identification of atypical data in FM's median it is considered a 5% quantile and the curves are identified as 2 and 3.

Table IV summarizes functional data identified as atypical, considering each depth measurement and each sample.

TABLE IV
PERCENTAGE OF CORRECT ANSWERS IN TUMORS CLASSIFICATION

Depth	Total Sample	Normal Tumor	Malignat Tumor
Mode	2.2	2.2	40
FM	2.3	2.3	40
Rp	2.2	2.2	40.5

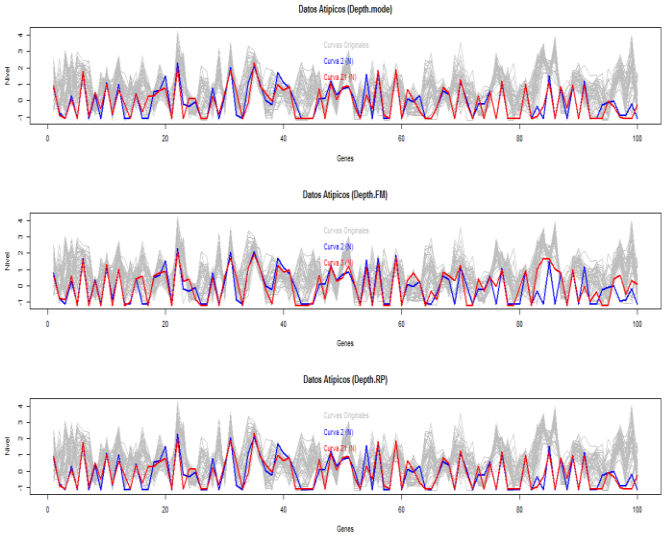

Figure 9: Curves identified as atypical functional data for each type of depth measurement.

In Figure 9, three graphs are shown. On each one, original curves are presented in gray and data identified as atypical in blue and red.

These results at first glance might indicate to us that there are only atypical data in normal tumors and that there are no atypical in malignant tumors, but performing the same analysis to identify atypical data in the sample of malignant tumors, it comes down to detect as atypical curves 40 and 48.

While in the subsample of malignant tumors curves 40 and 41 are identified as outlier. It is recalled that in the " Distance between functional data " section, in applying the distance by principal components to make a first approach to a classification rule, could not be correctly classify the curve 40.

V. Generalized Functional Linear Model (GFLM)

This section provides a Generalized Linear Functional Model (GFLM) where the functional covariates are: the level of expression of genes denoted as: **X=X (t)**, and, the first **X'(t)** and its second derivative **X''(t)** denoted as $\mathbf{X_1}$ and $\mathbf{X_2}$, respectively; and as response scalar variable (binary) the cancer type denoted as **Y** (0 = normal tumor, 1 = malignant tumor).

In this case, as the GFLM works with a binary response variable, this model provides a classification rule for the type of cancer (Bayes' rule).

This model is also called Functional Logistic Regression (Febrero-Bande, M. and Gonzalez-Manteiga, W. 2012), i.e. the models explain the relationship between **Y** (binary response) and a functional covariate **X (t)** by base representation X (t) and β (t). The functional model of logistic regression of the probability $\boldsymbol{\pi_i}$, the occurrence of an event, $\mathbf{Y_i} = 1$, rather than $\mathbf{Y_i} = 0$, conditioned on a vector of **covariates** $\mathbf{X_i}$ **(t)** is expressed as :

$$y_i = \pi_i + \epsilon_i \ , i = 1, \ldots, n$$

TABLE V
PERCENTAGE OF CORRECT CLASSIFICATION OF TUMORS

Number	Model	AIC	% classification
1	Y~X	49.7	82
2	Y~X_1	21.8	96
3	Y~X_2	24.1	96
4	Y~X+X_1	22.0	100
5	Y~X+X_2	22.0	100
6	Y~X_1+X_2	22.0	100
7	Y~X+X_1+X_2	32.0	100

Where $\boldsymbol{\pi_i}$ is the expectation of **Y** given $\mathbf{X_i}$ **(t)** modeled as follows:

$$\pi_i = P[Y = 1|x_i(t) : t \in T] = \frac{exp\left\{\int_T X_i(t)\beta(t)dt\right\}}{1 + exp\left\{\int_T X_i(t)\beta(t)dt\right\}} \quad , i = 1, \ldots, n$$

With $\boldsymbol{\epsilon_i}$ as independent errors with mean zero.

The functional variables used to estimate the model are:

1) Y = binary variable that identifies the type of tumor (0 = normal tumor, 1 = malignant tumor)
2) X = expression level of 100 genes of each individual
3) X1 = first derivative of the expression level of 100 genes of each individual
4) X2 = second derivative of the expression level of 100 genes of each individual

From these variables, they were estimated and compared seven models, Table III summarizes the characteristics evaluated to select the best model to use for the classification of tumor types. It is worth mentioning that the "fregre.glm" function from the R fda.usc package was used and B - Spline as representation basis for the seven models.

Additionally, it was explored with a representation based on principal component (PLS) for models 1 and 2 (it was used an R code for this) to improve results in adjustment and classification; but the results are similar to the representation in B - SPLINE therefore not proceeded to make estimates with this type of representation based .

The criteria used are: AIC (while lower is better); the percentage of tumors that are classified correctly from the total sample (% classification); and the percentage of prediction, that is, the percentage of tumors that are classified correctly from the total test sample (% Prediction). To calculate the prediction percentage, 10 test samples were used, 5 of normal tumors and 5 of malignant tumors; these were taken randomly setting a seed.

In the first model (see Table V), only are considered the original data (levels of gene expression), this is the model that explain less (AIC = 49.7) and its classification and prediction percentages are 82 % and 80 % respectively; on the other hand, with respect to the significance of the model parameters, we have that the first component (ab.bspl4.1) is significant (0.00639) to a level of significance of 5%. All parameters for the other models are not statistically significant at a level of significance lower than 1 %.

The second model (see Table V), has the lowest AIC (21.8) of all the proposed models, but there are models with better percentages of classification and prediction. From Model 4 to Model 7, the percentage of classification and prediction is 100 %, except for model 6 which has only a 70 % of prediction.

In general, when only are considered single-variable models of explanation for the type of cancer; AIC coefficient, the percentage of classification and prediction are the worst of all the proposed models.

Furthermore, it can be seen in Table V that increasing the number of variables in a model, the classification percentage improves up to 100%; however, when only considered in model 6, the functional variables: first and second derivative, the percentage of prediction is 70 %; and, when you have a more complex model with three functional variables considering the original data, its first and second derivative prediction, the model improves prediction but worsens explanation (best fit); In conclusion, one has that the complex model is good for predicting but not to explain the behavior of the cancer type variable.

In Table V, the painted yellow rows indicate the two models that have the same characteristics of explanation (best fit), classification and prediction; models 4 and 5 are the best models of seven models estimated. Therefore, the best model to classify tumors in normal and malignant is that which consider original data and one additional functional variable which can be the first or second derivative of the original data; this model comes to have a classification and prediction efficiency of 100%.

If the classification results obtained with models 4 and 5 are compared with the classification procedure by means of the distances between functional data used in section three which showed an efficiency of 98 %, we could say that for this sample, a Functional Generalized Linear Model is (GFLM) is more robust to the presence of outliers, as it allows an classification and prediction efficiency of 100%.

Finally, to complete this work Figure 10 shows the adjustments of the GLFM models for the cancer type variable when the entire sample of tumors is considered. It is worth mentioning that the graph settings for models of more than one explanatory functional variable is equal for all, because from model 4 to model 7 all have a classification percentage of 100%.

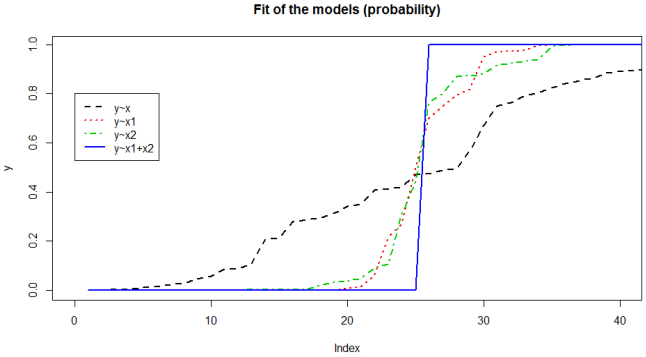

Figure 10: Functional Mean for original data and second derivative for each type of tumor.

VI. Conclusions And Future Research

1) It has greater discrimination when working with the first and second derivative of the expression of genes. This is reflected also in calculating the functional mean and variance in these spaces.

2) In the section "Outliers Detection", curves 2,3,21 and 40 are determined as outliers. These are not classified correctly using the cluster method but by using the functional generalized linear model.

3) Increasing the number of variables in the functional generalized linear model, the classification percentage improves up to 100%. The functional variables included were the first and second derivative.

4) The functional data analysis is very recent in the fields of statistics and medicine, despite this there is an increased interest in using this methodology. It is intended to continue to address problems of classification in other areas of science.

5) Specifically for the medical field, will work to make a functional generalized additive linear model that eliminates the restriction of linearity for the independent variables.

6) Besides, it is addressing functional models to describe the relationship of the expressions of genes with other variables related to cancer.

References

[1] Cuevas A, Febrero M, Fraiman R. 2001. Cluster Analysis: a further approach based on density estimation. Computational Statisticsand Data Analysis 36: 441–456.

[2] Dudoit et al. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data, Journal of the American Statistical Association, 97 (457), 77-87.

[3] Febrero-Bande, M. and Oviedo de la Fuente, M. 2012. Statistical computing in functional data analysis: The R package fda.usc. Journal of Statistical Software, 51(4):1-28.

[4] Febrero-Bande, M. and Gonzalez-Manteiga, W. 2012. Generalized additive models for functional data. TEST, 22(2):278-292.

[5] Ferraty, F. and Vieu, P. 2006. “Nonparametric Functional Data Analysis: Theory” and Practice. Springer-Verlag, New York., Pp. 113-146.

[6] Fraiman R. and Muniz G. 2001 Trimmed means for functional data, Test, 10(2), 419-440.

[7] Lopez-Pintado, S., Romo, J., Torrente A. 2010. “Robust depth-based tool for the analysis of gene expression data”. Biostatistics 11, 2, pp 254-264.

[8] Ramsay, J. O. and Silverman, B. W.2005. “Functional Data Analysis”, 2nd ed., Springer-Verlag, New York., pp. 147-325.

[9] Romualdi C., Campanaro S., Campagna D., Celegato B., Cannata,N, Toppo S.,Valle G. and Lanfranchi G. 2003 Pattern recognition in gene expression profiling using DNA array: a comparative study of different statistical methods applied to cancer classification. Human Molecular Genetics **12**, 823-836.

[10] Singh D. et al. 2002. Gene expression correlates of clinincal prostate cancer behavior, Cancer cell, 1 (2), 203-209.

[11] Tárraga J., Medina I., Carbonell J., Huerta-Cepas J., Mínguez P., Alloza E., Al-Shahrour F., Vegas-Azcarate S. Gotz S. Escobar P and others 2008. GEPAS a web-based tool for microarray data analysis and interpretation. Nucleic Acids Research 36, W308-W314.

[12] Wessels L.F.A., Reinders M. J. T., Hart,A.A.M.,Veenman C.J., Dai H., He Y.D. and Van’t Veer L.J. 2005. A protocol for building and evaluating predictors of disease state based on microarray data. Bioinformatics 21, 3755-3762.

[13] Zuo Y, Serfling R. 2000. General notions of statistical depth function. Annals of Statistics 28: 461–482.

Miguel Flores, is a professor at the National Polytechnic School and a researcher at the Center for Modeling Mathematics at the National Polytechnic School in Quito, Ecuador. He is a BSc. in Statistical Computing Engineer from the Polytechnic School of the Coast. In 2006 he received an in MSc. in Operations Research from the National Polytechnic School, and in 2013 received a MSc. in Technical Statistics from the University of A Coruña. He is currently a doctoral student at the University of A Coruña in the area of Statistics and Operations Research. He has over 15 years professional experience in various areas of Statistics, Computing and Optimization, multivariate data analysis, econometric, Market Research, Quality Control, definition and construction of systems indicators, development of applications and optimization modeling.
ORCID ID: 0000-0002-7742-1247

Guido Saltos S. received his Engineering degree in Electronics from Escuela Politécnica del Ejército (ESPE), Quito, Ecuador in 1987. He received his M.S. degree in Applied Statistics from National Polytechnic School (EPN), Quito, Ecuador, in 2016. He worked several years in the field of industrial automation and now he is working at Universidad de las Américas (UDLA) in Quito Ecuador. His interests are related with data depth, and non-parametric statistics.

Sergio Castillo Páez, is a mathematical engineer graduated from the National Polytechnic School in 2002. He also studied finance in the Simon Bolivar Andean University, and is currently studying his PhD in Statistics at the University of Vigo, Spain. He is a professor at the ESPE Armed Forces University in Ecuador. His current lines of research are related to geostatistics and analysis of multivariate data.

A fresh recipe for designers: HCI approach to explore the nexus between design techniques and formal methods in software development

Julián Andrés Galindo Losada

***Abstract*—Emerging companies involved in design and implementation of innovative products demand multidisciplinary teams to be competitive in the market. This need mainly exposes designers to extend their knowledge not only in User Interface elements of the design process but also in software methodologies to cover the lack of resources and expertise in start-ups. It raises the question of how designers can line up HCI techniques with best practices in software development while preserving usability and easy-to-use principles. To explore this gap, this paper proposes an approach which combines existing technology and methods by studying the nexus between HCI prototyping and software engineering. The approach is applied into a case study in the design of a virtual shop harmonizing the use of storyboards and the spiral. A comprehensive analysis is performed by using a Technology acceptance model (TAM) regarding with two variables:usability and easy-to-use. The present finding underlines the positive integration of HCI techniques and formal methods without compromising user satisfaction with a potential benefit for small companies in a formation stage.**

***Index Terms*—human computer interaction, software methodologies, virtual shop, shopping on-line, information systems**

I. Introduction

To date, entrepreneurial ventures as design-center enterprises aim to a rapid prototyping of products in a scalable and sustainable approach [13] where during their formation stage there is no a correct skills balance in the workforce[4]. This fuzzy beginning leads to an ineffable pressure on designers to extend their knowledge to cover other software development areas to deliver quality products such as requirement analysis, implementation, testing and evaluation.

Particularly, designers are generally trained in user interface elements such as ergonomics, interaction design, multimedia, content [3] as well as usability principles such as learnability, operability and attractiveness which may causes a lack of experience in software development methodologies. Furthermore, current designing tools might not provide enough support to designers to have a complete understanding of the software process. In this context, one may be interested to know the feasibility of integration between Human-Computer Interaction (HCI) techniques and software methodologies so that designers can learn from best software practices while preserving HCI principles.

Julián Galindo is currently an active researcher of the *"Laboratoire d'informatique de Grenoble", LIG, Grenoble Alps University, UGA, Grenoble, France in the domain of UI adaptation driven by user emotions, julian.galindo@imag.fr - juliangalindo.com/research.*

Our contribution is a method to improve existing approaches by unifying a HCI process (storyboards) and a software development model (spiral) to enrich the capabilities of designers during the design process. The method is applied in the designing of a virtual shop and its evaluation is reported by using user testing reports which is also compared with the Technology Acceptance Model [7].The finding is positioned by Action Research (AR) as high application domain and low solution maturity, in which the results of studying the integration in HCI and software development aiming to solve an existing issue in a particular social setting[1].Overall, the evidence suggests that a design technique can be aligned with a software method; notably, when both are inherent iterative processes. Furthermore, the method's application result in a product with valuable degrees of usability and easy-to-use features. A priceless rewarding for designers and end-users.

The reminder of the paper starts by explaining the coverage of HCI and software development, related work, approach in action, prototype description and evaluation.

II. Background in HCI and software development

HCI can be defined as the process of analysis and design of interfaces between humans and computers [9], in others words the analysis and design of users interfaces. The process of design interfaces with HCI can be done iteratively with prototypes [6]. The use of prototypes allows to test early versions of systems with real users. A technique of prototyping are storyboards. A storyboard is a graphical depiction of the outward appearance of the intended system, without any accompanying system functionality. Although a storyboards technique provides more flexibility to the designer to focus on users tasks and UI graphical elements, there is a formal software methodology to complement the design process of prototypes such as rapid prototyping and the spiral model of software development.

A spiral model of software development involves a process expressed in a four-cycle plan [2]: define objectives, alternatives and constrains; evaluate alternatives, identify and resolve risks; develop and verify next-level product; and finally plan next product's phases. Hence, if a HCI process with a software model are defined in conjoint as process' techniques, it may be applied to improve the process of designing a virtual shopping interface.

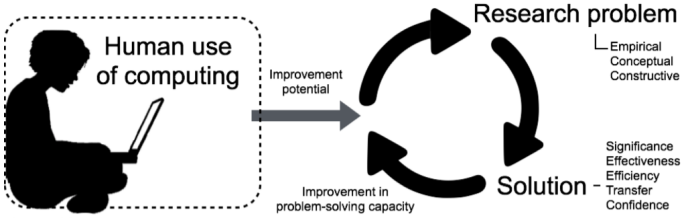

Fig. 1: HCI research as problem-solving

The role of HCI design has been explained by introducing the term "design" meant usability engineering as the the process of modeling users and systems and specifying system behavior such that it fitted the users tasks, was efficient, easy to use and easy to learn. [21]. During the design process, designers work with software engineers where this synergy leads to a set of useful skills such as navigation, typography, topography, visual hierarchy, color which is underlined as a creative design [23]. A process which is also different from an engineering approach. It is mainly because in a creative design, designers make a continuous process to fit user tasks by to looking and questioning at present the problematic while developers focus more on software specifications[8].

In terms of research methods and the ability to solve problems in human computing, HCI can be characterized as a problem-solving field (Fig. 1) [10, p. 5]. It involves "1) subject of inquiry, human use of computing; 2) research problems; 3) types of problem-solving capacity pursued; and 4) achievements in improving problem-solving capacity".

The designer invest time in understand the problematic as well as human and computer interaction. Later, this understanding causes la definition of the research problem divided in an Empirical(unknown phenomena), Conceptual(implausibility) and Constructive approach(no known solution).In this point, the designer can decide what kind of research problem is facing to provide a better solution; this solution defines heuristics aspects for contributing and assessing to the capacity's evolution in problem-solving such as significance, effectiveness, efficiency, transfer and confidence [10, p. 4].

Similarly, the concept of problem-solving in HCI can also be seen in Information systems as a research method known as Action Research (AR). It covers the result of scientific knowledge trough the study of the effects of an action taken which aims to solve an existing issue in a specific social environment [7].

Once, the role of HCI in research has been explored, it is necessary to focus on a HCI technique to solve human-computer issues. The storyboards' technique is described as a logical process which aims to describe the interaction between the user and the system over the time through the use of graphical elements - often sketches, and textual narrative- to reduce the cost of designing in a cyclical process to achieve user goals [6]. The storyboard process includes a two iterative process; make a design and then test and collect data (Fig. 1).

First, the designer makes a design regarding with the user needs. Second, the designer evaluates the design, testing it with the user to understand the constrains and future improvements. All information is collecting by using formal interviews where the user expresses his level of prototype acceptance. Finally, the designer and user repeats the same process until the prototype meets user requirements.

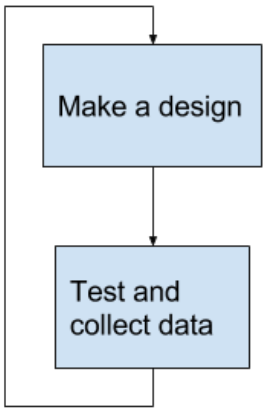

Fig. 2: Storyboard process

This technique can be seen as an iterative helpful tool for rapid prototyping (RP) in the design process where many iterations are needed until the solution meets the user needs. Consequently, this concept can be extended by using the Information Systems (IS) prototyping method in Action research (AR) that includes not only the prototype design but also its evaluation in a cyclical process [7] (Fig. 2 and 3).

The mentioned process includes the following steps:

1) Diagnosing. To identify the main issues.
2) Action planning. To specify actions to solve or improve.
3) Action taking. To implement the action planned by an active intervention in the participating organizations.
4) Evaluating. To compile the evaluation of the results by study subjects and researchers.
5) Specifying learning. To identify new knowledge.

On the other hand, software engineering provides a cyclical methodology based on prototypes which is called the spiral model. A spiral model includes four steps: determine objectives, identify risks, development and testing, and finally plan the next iteration [2, p. 2].

Boehm states that the usage of this model is possible with one table per iteration with includes objectives, constraints, alternatives, risks, risk resolution, risk resolution results, plan for next phase and commitment. For instance, table 1 shows an example of the first iteration of this model for a TRW Software Productivity System.

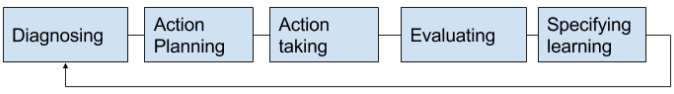

Fig. 3: IS prototyping model

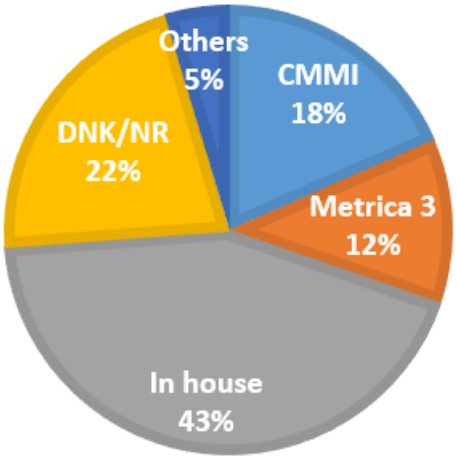

Fig. 4: Main software methodologies

Designers need to know how users perceive systems or prototypes in order to understand how easy or difficult is the use of one particular technology. Many different models have been introduced in the research field of Information systems to evaluate the level of technology acceptance. In fact, the technology acceptance model (TAM) by Davis, Venkatesh [5] which aims to measure the degree of perceived usefulness and ease-of-use that a user has when using or interacting with some particular technology. As it is shown in Fig. 6, there are many developed models such as TAM2 [17], TAM3 [16], the Unified Theory of Acceptance and Use of Technology (UTAUT) [18], and UTAUT2 [19].

Every variation of the TAM model implies itself the user's behavioral intention against any technology. In detail, to understand the value of user behavior in TAM, two variables (usefulness and easy-of-use) are combined to define the user attitude which leads to the determination of the behaviors' degree. Then, this degree can infer the actual use of the technology. To extend it, a more precise forces' explanation which influence the perceived value of usefulness was given in TAM2; whereas, TAM3 focus on the easy-to-use perception. Similarly, an understanding of how the variables change over time was introduced in UTAUT, and its improvement UTAUT2.

III. Related work

This sections reviews the HCI literature to understand historical, highlights and proposed solutions and more specifically, interactive systems regarding with HCI design methodologies and virtual shop designs.

The figure 4 evidences that enterprises prefer developing their own methodologies (in house) [12], mainly because high complexity and previous know-how for adopting the technology as luxury items that will now allow small companies to be competitive.

A well-known technology, UML is found with less than 1 per cent. Despite UML is recognized as a standard unified modeling language, it has been criticized for a lack of notations and inconsistencies in semantic which impacts the learning curve for designers[15]. Even more, this technology might be a straitjacket for standalone enterprises because a rigid structure will rarely promote creativity and innovation being a vital element for small companies. Designers can not also be aware of testing and technology adoption of final products by end-users unless a formal evaluation method is included in UML.

Regarding with virtual shop designs, there have been some attempts to design a virtual shopping system such as the Design and Implementation of a Collaborative Virtual Shopping System. It considers a multi-agent support for collaboration activities including simulation and interaction processes to covers a more realistic shopping experience [22]. Furthermore, it was developed by using VRML systems, intelligent software and network technologies. Similarly, a vCOM and 3D system [14] that allows users to navigate in a virtual e-commerce world while interacting with real time agents. Puglia presented a multi e-commerce solution by the interaction of multiple users participating in a simultaneous virtual shop experience [11]. More closely, HCI has shown some research projects such as the virtual store layout with three layout components free-form, grid, and racetrack [20].

IV. Approach in Action

This sections aims to explore the design overview, market relevance and implementation. Overall, the designer follows the definition or steps of the spiral model seen in previous sections; then, he uses storyboards to create the prototypes and evaluate the design versions with end-users. Consequently, this evaluation will be extended by showing a technology adoption model.

A. Overview

It was designed a prototype to test the concept of a virtual shop where the customer can interact by using a tablet and QR codes. As it was shown before, a spiral model is used as a software model containing as core component all other techniques such as the HCI storyboard technique (HST) and the TAM (technology acceptance model). The HST is used in every spiral iteration in the designing phase, it means that the designer focus on the UI elements of the interface by designing stories to understand the interaction between the user and the virtual shop. It is expected to reduce the complexity of the designing and the understanding of user behavior in the virtual shop usage. Then, the user tests the prototype and all information is collected by using surveys format, asking specific questions to clarify the requirements for the next prototype design. In the final prototype(release), the leader team will use a TAM approach to measure the level of technology adoption by the user.

It is expected that the HCI principles and techniques used in the designing phase will impact positively the user perception in usability and easy-to-use variables. The virtual shop

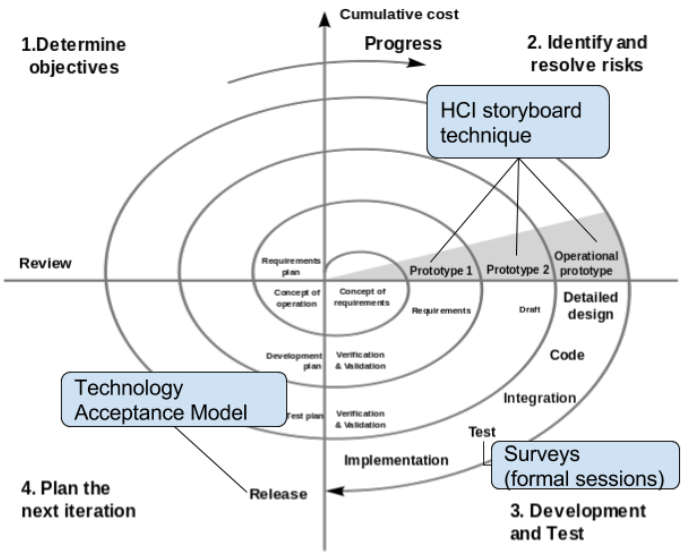

Fig. 5: Spiral model in the virtual shop design

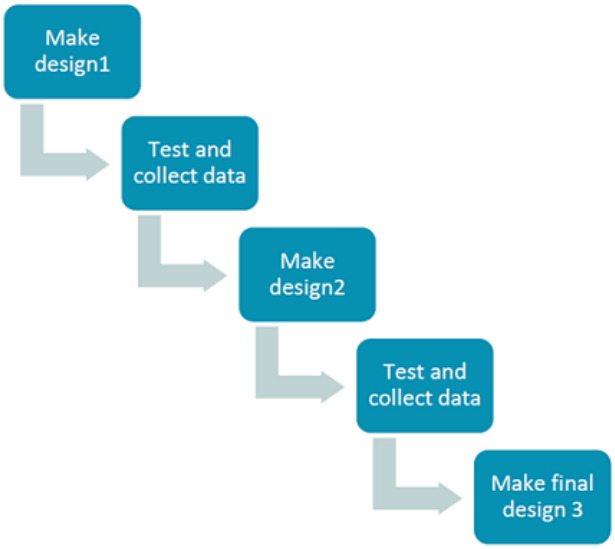

Fig. 7: Process to test the prototype

prototype supports these functions: select products, create an account, user login, do the payment and the delivery process. We are using the Woolworth's logo to show a familiar brand to the user.
The prototype focuses on two tasks:

1) Select three products and remove one product
2) Do the Checkout process (delivery info + payment + confirmation)

The prototype has two main interactive tasks:

1) The selection of products by using a products wall
2) The data entry of the purchase details

B. Market relevance

It is an opportunity to test what big supermarkets in Australia are doing with new interaction methods to buy products. Woolworths has introduced the new virtual shop in 2012 which is working in Sydney. In addition, the demand of buying on-line is increasing in Australia. Customer behaviors are changing so that big supermarkets like Woolworths are making new ways to interact with the client as a Business to Customer model.

Fig. 6: Prototype

C. Implementation

We made a prototype by using HTML, CSS and MySQL to publish the shop in our domain Based on the two tasks described previously, we followed this prototype plan:

1) Check the tasks requirements
2) Design the html interfaces.
3) Insert the products catalog with the most common products for students.
4) Publish the prototype.

The Fig. 7 shows the process to test the prototype.

The next subsections show the features that every design included which clarifies the main elements per iteration.

1) Design 1:

Spiral iteration 0
To illustrate, a comparison table of iterations was made [2].
Included

- Select products and do the payment on line by using the tablet.

Not Included

- No navegation
- No instructions in the product's wall
- No buttons update & remove
- No prices
- No biometric login by eyes
- No paypal
- No email product description
- No best/last/ products

[2]http://juliangalindo.com/hci/espiralmethod/modeliterations.png

2) Design 2:

Spiral iteration 1

To illustrate, a comparison table of iterations was made (previously shown).

Included

- Select products by using QR codes on the products wall
- navegation
- Instructions in the product's wall
- Buttons update & remove
- Biometric login by eyes
- paypal
- Email product description
- best/last/ products

Not Included

- No prices
- No interactive videos about products

3) Design 3:

Spiral iteration 2

To illustrate, a comparison table of iterations was made (previously shown).

Included

- Prices
- Interactive videos about products

D. Usability and user experience goals

We focused on how users can interact with a new way of shopping so that we set these goals:

- How do users feel when they select products by scanning QR codes?
- How do users feel when they see products without prices in a WALL of products?
- Do users prefer selecting products by interacting with a tablet?
- Is the checkout process difficult?

Therefore, we set these goals:

Tablet interaction

1) Selection of products by interacting with a tablet
2) Selection of products by interacting with a products wall by scanning QR codes.
3) Change the products quantity in the shopping cart by interacting with a tablet
4) Do the checkout process including three steps; (delivery info, payment and confirmation) by interacting with a tablet.

Products wall

5) Look for products with and without prices showed in the wall.
6) Look for products with QR codes showed in the wall.

V. Storyboard description

A. Stage 1

In this stage a user wants to scan for a product in order to know about a description, price, or to buy an item. QR codes are printed below each product .

B. Stage 2

In this stage a user scanned an item and wants to see many other products. Also, the virtual shop will offer alternative products in order to give other suggestions to the consumer.

C. Stage 3

Once a product is scanned, the cart from virtual shop show the quantity, the price of the item selected through QR codes.

D. Stage 4

Sometimes, a customer will want to update the cart, for example, to add or remove the quantity of items and then proceed to the checkout .

E. Stage 5

The goal of this new type of service is to deliver the product as soon as possible the checkout is completed, so the customer will receive its purchase at home.

VI. Prototype description

A. How did you make it?

We use the following technologies:

- **CMS:** osCommerce to create the prototype (template and web pages) as the most suitable tool according to the storyboards and user requirements.
- **QR CODES:** We use an on-line generator tool to create the QR codes.
- **Tablet emulator:** We emulate a tablet access by using Tablet Emulator.
- **Physical device:** We use a tablet with an android system.

B. Designs

This section shows the three designs with their respective descriptions for every planned task.

Design 1

Fig. 8: QR code inclusion in design 2

Fig. 9: QR code recognition in design 3

Task 1: Select 3 products and remove 1 product by using a tablet.

Steps:

1) Click in dairy products: The system shows all the products with images and the names .
2) Click in the milks image; the system shows more details about the product (price, description, and image) .
3) Click in add to cart button: the product is added to the shopping cart .
4) Selecting of the second product by following the same steps.
5) Selecting of the third product by following the same steps .
6) Deleting the last product.

Task 2: Do the Checkout process (delivery info + payment + confirmation).
Steps:

1) Click in the checkout button .
2) The user needs to login by creating an account and later on input the email and password.
3) The user needs to input information about delivery and payment .
4) After the user click in the continue button, a confirmation page is showed .
5) The user gets a confirmation email .

Design 2

This section will show the specific improvements between the design 1 and design 2.

1) The user could choose his products by using QR codes which are printed in a products wall (Fig. 8).
2) There is a navigation menu for each product page
3) The login supports a biometric process (eyes login)
4) The checkout process supports PayPal
5) Special and popular products are showed in the right block
6) More details were included in the emails confirmation

Design 3

This section will show the final improvements between the design 2 and the design 3.

1) A product video was included.
2) Prices were included.
3) The user scans the QR codes by using its tablet (Fig. 9).
4) The selected product is loaded in the tablet automatically.

VII. First User testing plan

A. Describe the procedure of the user test

The user will be using the QR codes printed below a product; the main purpose is to scan them in order to buy the item or just to know the price of the scanned item. Hence, the satisfaction of the user is going to be analyzed through surveys.

B. The tasks to be use in the test

- Scanning printed products in the wall.
- Flexibility to buy an item.
- Payments on line.
- Procedure to add/remove items.

C. Who will be the participants?

There will be volunteers that tested this new innovated way to buy items. Volunteers were people representing a typical user that leaves work and go back home, but they do not have enough time to buy groceries.

D. How data will be collected and record?

The main from to collect data is through QR codes, and these codes redirect the user to a virtual store that process the orders and record them in each account. In order to measure the user satisfaction, a survey was conducted to see the results according to the activities done when the volunteers bought an item.

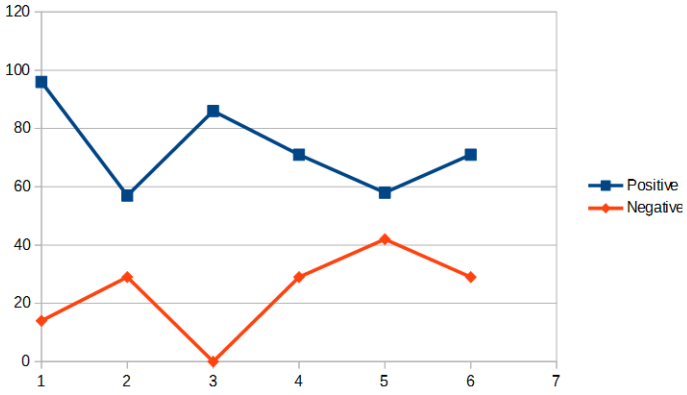

Fig. 10: TAM usability acceptance

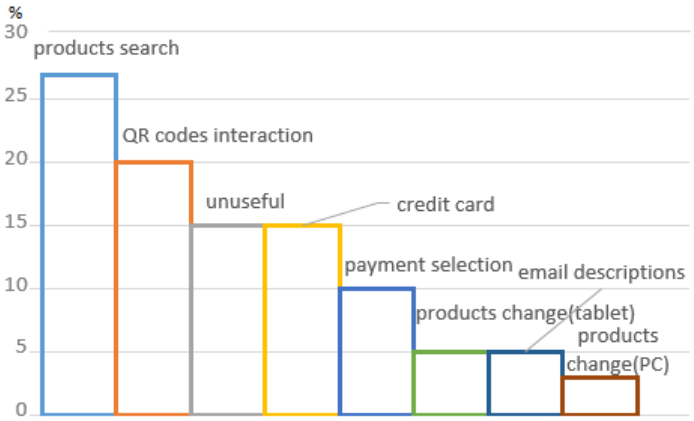

Fig. 11: TAM easy-to-use acceptance

E. Script for participant briefing and demonstrations

1) A user sees the wall where the products are printed.
2) With his tablet / mobile, scan the QR code and the rice is displayed on the screen.
3) If the user wants to buy, he can click on add to cart.
4) Then the application precedes to checkout and delivery the product.

VIII. First User testing report

A. What is the testing environment?

A testing environment is a setup of software and hardware to perform a product. The environment used for this test was driven in a closed scenario simulating a bus stop station due that many people that do not have enough time to buy groceries this will be an alternative to improve their lifes quality.

B. Testing results

With the first survey, we understood that users prefer eye biometric method to login, want a description of the products in their email, want to see best-latest products and receive suggestions of other products in order to have a better combination.

C. How the results may help to refine the designs

With the results shown above, the second design will be improved according those users feelings about the procedures taken using QR codes. In addition, some improvements will be included in the second design.

IX. Second User testing plan

A. Describe the procedure of the user test

Once the user used his device to scan a QR code, now the measures are about the usability of the Virtual Shop. The tests are focused in the procedure of buying an item on the wall, the users satisfaction about the use and comfort of the usability of the process.

B. The tasks to be use in the test

- Scanning printed products in the wall.
- Flexibility to buy an item.
- Payments on line.
- Procedure to add/remove items.
- Checkout procedures.

C. Who will be the participants?

There will be volunteers tested this new innovated way to buy items. Volunteers were people representing a typical user that leaves work and go back home, but they do not have enough time to buy groceries in a store, so the target of this project are people who wants to buy groceries stuff and receive them at home.

D. How data will be collected and record?

The main from to collect data is through QR codes, and these codes redirect the user to a virtual store that process the orders and record them in each account. In order to measure the user satisfaction, a survey was conducted to see the results according to the activities done when the volunteers bought an item.

E. Script for participant briefing and demonstrations

1) A user sees the wall where the products are printed.
2) With his tablet / mobile, scan QR code and the rice is displayed on the screen.
3) If the user wants to buy, he can click on add button and add to the cart.
4) Additional security to login in the Virtual Shop is using the eye using biometric algorithms in the application.
5) Then the application precedes to checkout and delivery the product.

X. Second User testing report

A. Where is the testing environment taken?

Habitually, bus stations are crowded most part of the daytime, these tests were conducted in a similar situation, and the library was the indicated place to recollect data and also to test the performance and usability of the prototype.

B. Who are the participants

Participants are the same group of users that used the first version of the Virtual Store in the same conditions but with more characteristics in the application. In addition, some others joined to the test for the first time in order to give us a feedback about the virtual shop.

C. Testing results

With the second survey (See Appendix) we understood that users consider secure and fast eye biometric method to login, consider useful the description of each product purchased is sent to the email and prefer a video to have a clear description of an item.

D. How the results may help to refine the designs

It is clear, according to the chart that users are more satisfied with the previous version due its easier way to modify the cart from the Virtual Shop. In addition, in the second version is added a biometric login with the human eye, it is safer and faster. However, the volunteers said that sometimes felt the need of a better explanation of an item. Sometimes new products can be released and no one can explain a specific item, so the new version of the Virtual Shop will be created in order to satisfy the users feedback.

E. Evaluation

By following the TAM model, the team leader performed the usability, easy-to-use and action use by asking the user its experience with the final prototype and collecting the information. The following charts shows how high or low are the user perception in every variable:usability and easy-to-use. In this preliminary evaluation, users showed more positive evidence to adopt the technology across all user tests (Fig.10). In fact, users found a considerable degree of easy-to-use mainly at interacting with QR codes and looking for products (Fig.11).

XI. Discussion and Conclusion

Based on the degree of usability and easy-to-use of the final design, it is possible to suggest that a formal software method can work in harmony with a HCI technique like storyboards. Designers will take advantage of this approach or method to extend their capabilities in best practices in software development as well as design techniques. Overall, this integration was mainly evidenced because both techniques use a iterative model to achieve product goals. In storyboards, every stage was improved regarding with the previous version while the spiral model outfit a formal method where the designer creates every run to clarify and formalize all steps of the design process. Particularly, this method is highly recommended to startups in their formation stage where multiple roles demand synergy in designers to overpass lack of knowledge in the design process.

To follow a software methodology like the spiral model provides a more formal set of activities where team members can know exactly what to do and what the impacts are or potential risks in the interface for every prototype. Furthermore,HCI provides a helpful and easy technique to use where designers and users can meet their needs spending time in prototyping instead of developing. User testing reports helped designers to understand what are the most valuable requirements for the user while TAM releases a better understanding of user behavior. The assumption, ”user will not change his behavior so that the usability of the technology will remains” can affect considerably the use of the virtual shop (final prototype) over time. It means that more considerations must be taken in order to run TAM iterations more frequently;for instance, while the user changes his perception the technology does too.

In spite of showing a positive technology adoption in a formation stage of a company, this research might be extended to cover other main aspects. For instance, the use of agile methodologies in conjunction with other HCI techniques. The study of the correlation between cost and benefit of using formal methods by designers. A more strict review of the impact in start-ups during the formation stage as a input to the next stage ”Validation”. to understand the degree of growth acquired. This research followed the assumption ”designers are proactive employees to undertake new technology challenges”; however, the barrier or learning curve in formal methods may lead to a lack of enterprise resources.

Appendix A
Results of first survey

How often would you buy an item through your phone?

- 1 0 (0%)
- 2 0 (0%)
- 3 1 (14%)
- 4 4 (57%)
- 5 2 (29%)

Where do you want to see the price of an item?

- Below the product 2 (29%)
- Through QR Codes 5 (71%)
- Through a web page 0 (0%)

Would you need a guide in your shopping cart in order to see more products?

- Yes 6 (86%)
- No 1 (14%)

Do you agree to have a set of detailed instructions in order to guide to buy an item?

- 1 0 (0%)
- 2 1 (14%)
- 3 0 (0%)
- 4 2 (29%)
- 5 4 (57%)

How do you prefer to modify your cart to add or remove an item in your shopping cart?

- Labelled ”Add” and ”Remove” buttons 6 (86%)
- + / - symbols 1 (14%)
- Modifying directly quantity 0 (0%)

How do you prefer to pay the selected item?

- Credit Card 1 (14%)
- PayPal 4 (57%)
- Debit Card 1 (14%)
- Prepaid Card 1 (14%)

How do you prefer to login to the virtual store?

- Username / password 0 (0%)
- Eye biometric method 5 (71%)
- Email / password 2 (29%)
- Fingerprint biometric method 0 (0%)

Do you prefer a description of the products in your email?

- Yes 6 (86%)
- No 0 (0%)
- Choose at the moment of checkout 1 (14%)

What do you prefer to see meanwhile you buy an item?

- Best Product 1 (14%)
- Latest Product 1 (14%)
- Best - Latest products 4 (57%)
- Anything 1 (14%)

APPENDIX B
RESULTS OF SECOND SURVEY

It is easier for you to buy an item using QR codes?

- 1 5 (71%)
- 2 2 (29%)
- 3 0 (0%)
- 4 0 (0%)

Seems useful navigation bar in the application?

- Yes 4 (57%)
- No 2 (29%)
- Not important 1 (14%)

APPENDIX C
STORYBOARDS PER STAGE AND SPIRAL ITERATION

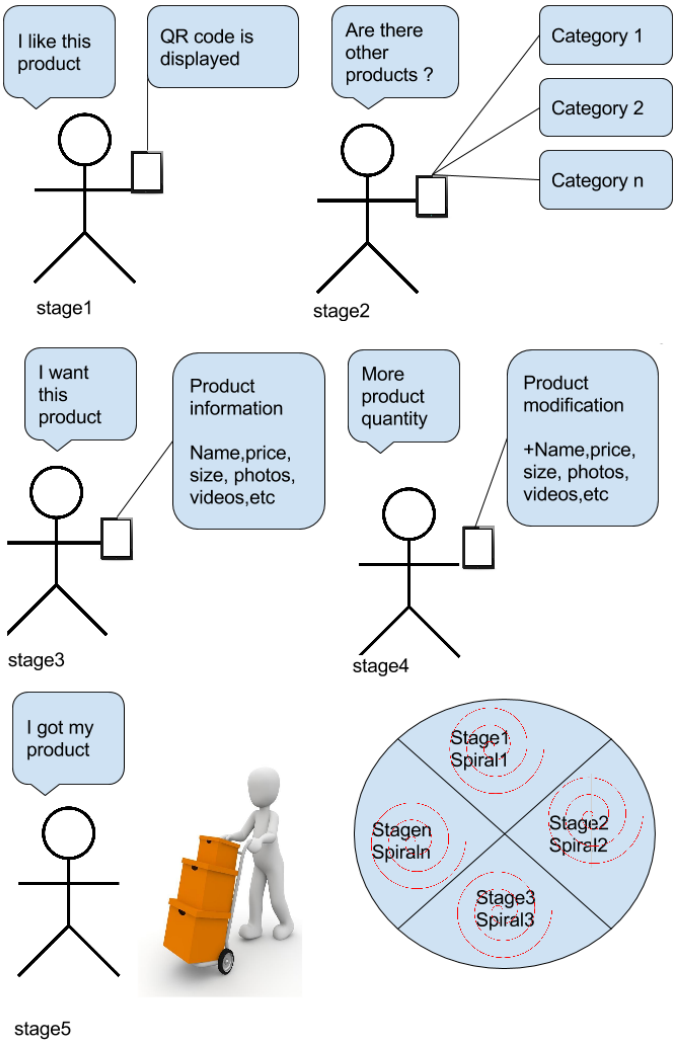

Fig. 12: Storyboards per stage

ACKNOWLEDGMENT

The author wishes to thank God, reviewers and collaborators for their support, highlights, designs, data processing and quality time.

REFERENCES

[1] Richard Baskerville and A. Trevor Wood-Harper. Diversity in information systems action research methods. *European Journal of information systems*, 7(2):90–107, 1998.

[2] B Boehm. A Spiral Model of Software Development and Enhancement. *SIGSOFT Softw. Eng. Notes*, 11(4):14–24, August 1986.

[3] John M. Carroll. Human-computer interaction: psychology as a science of design. *Annual review of psychology*, 48(1):61–83, 1997.

[4] Selene Se Lui Chew. *Designers as Entrepreneurs: An Investigation on Why Startups Need Design and Design Need Startups*. PhD thesis, The Ohio State University, 2015.

[5] Fred D. Davis, Richard P. Bagozzi, and Paul R. Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Management science*, 35(8):982–1003, 1989.
[6] Alan Dix. *Human-computer interaction*. Springer, 2009.
[7] Edison Loza and Alex Buitrago. Qualitative assessment of user acceptance within Action Design Research and Action Research: two case studies. 2014.
[8] Jonas Lwgren. Applying design methodology to software development. In *Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*, pages 87–95. ACM, 1995.
[9] Mari-Carmen Marcos. HCI (human computer interaction): concepto y desarrollo. *El profesional de la informacin*, 10(6):4–16, 2001.
[10] Antti Oulasvirta and Kasper Hornb\a ek. HCI Research as Problem-Solving. 2016.
[11] Stefano Puglia, Robert Carter, and Ravi Jain. MultECommerce: A distributed architecture for collaborative shopping on the www. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pages 215–224. ACM, 2000.
[12] Colomo-Palacios Ricardo. *Agile Estimation Techniques and Innovative Approaches to Software Process Improvement*. IGI Global, February 2014. Google-Books-ID: CAOXBQAAQBAJ.
[13] Natalie Robehmed. What is a Startup. *Retrieved from site forbes: http://www. forbes. com/sites/natalierobehmed/2013/12/16/what-is-a-startup*, 2013.
[14] Xiaojun Shen, T. Radakrishnan, and Nicolas D. Georganas. vCOM: Electronic commerce in a collaborative virtual world. *Electronic Commerce Research and Applications*, 1(3):281–300, 2003.
[15] Keng Siau and Poi-Peng Loo. Identifying difficulties in learning UML. *Information Systems Management*, 23(3):43–51, 2006.
[16] Viswanath Venkatesh and Hillol Bala. Technology acceptance model 3 and a research agenda on interventions. *Decision sciences*, 39(2):273–315, 2008.
[17] Viswanath Venkatesh and Fred D. Davis. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science*, 46(2):186–204, 2000.
[18] Viswanath Venkatesh, Michael G. Morris, Gordon B. Davis, and Fred D. Davis. User acceptance of information technology: Toward a unified view. *MIS quarterly*, pages 425–478, 2003.
[19] Viswanath Venkatesh, James YL Thong, and Xin Xu. Consumer acceptance and use of information technology: extending the unified theory of acceptance and use of technology. *MIS quarterly*, 36(1):157–178, 2012.
[20] Adam P Vrechopoulos, Robert M OKeefe, Georgios I Doukidis, and George J Siomkos. Virtual store layout: an experimental comparison in the context of grocery retail. *Journal of Retailing*, 80(1):13–22, January 2004.
[21] Peter Wright, Mark Blythe, and John McCarthy. User experience and the idea of design in HCI. In *Interactive Systems. Design, Specification, and Verification*, pages 1–14. Springer, 2005.
[22] Lu Ye, Bing Xu, Qingge Ji, Zhigeng Pan, and Hongwei Yang. Design and Implementation of a Collaborative Virtual Shopping System. In Weiming Shen, Zongkai Lin, Jean-Paul A. Barths, and Tangqiu Li, editors, *Computer Supported Cooperative Work in Design I*, number 3168 in Lecture Notes in Computer Science, pages 309–318. Springer Berlin Heidelberg, May 2004. DOI: 10.1007/11568421_31.
[23] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. Research through design as a method for interaction design research in HCI. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 493–502. ACM, 2007.

Julian Galindo was born in Quito, Ecuador, in 1982. He received a bachelor degree in informatics engineering from Central University, UCE, Quito, Ecuador, in 2007. The Master in Information Technology from James Cook University, JCU, Townsville, Australia (2012). In 2014, he joined the Faculty of Engineering in Systems, National Polytechnic School, EPN, as a Professor. Since October 2016, he has been working with the "Laboratoire d'informatique de Grenoble", LIG, Grenoble Alps University, UGA, Grenoble, France as a PHD student in the domain of Human Computer Interfaces.

Estudio exploratorio de la técnicaTimming Attack en el criptosistema RSA
Exploratory study of Timing Attack on RSA cryptosystem

Francisco Bolaños Burgos, Luis García Tenesaca y Antonio Cevallos Gamboa

***Resumen*— El presente trabajo realiza un análisis bibliográfico exploratorio del tipo de ataque *Timing Attack (TA)* de *On The Side Channel Attack (SCA) en RSA*. Para lo cual, se analizaron los activos de información, los modos de operación y las contramedidas efectuadas de 22 artículos. Los resultados evidencian que el activo de información que más ataques tuvo son las tarjetas inteligentes (32%), la contramedida mayormente aplicada es el cegamiento (33%) y los modos de operación más utilizados son el Chinese Remainder Theorem (CRT) o Montgomery Multiplication (MM) con CRT (41%). Adicionalmente se evidencia que sólo un ataque fue realizado a los sistemas de telecomunicaciones, lo cual permite plantear trabajos futuros en el análisis de la misma técnica con base en las tecnologías WiMAX y el protocolo SIP de VoIP.**

***Palabras Claves*— Side Channel Attacks; Timing Attack; RSA;activos de información; modos de operación; contramedidas**

***Abstract*— This paper makes an exploratory bibliographic analysis of the Timing Attack (TA) technique on the Side Channel Attacks (SCA) in RSA. The information assets, operation modes and countermeasures of 22 papers were analyzed. Findings show that smartcards are the most attacked information assets (32%), blinding is the most applied countermeasure (33%) and the Chinese Remainder Theorem (CRT) or Montgomery Multiplication (MM) with CRT are the most frequent operation modes(41%). Furthermore, just one attack was executed in telecomunication systems, this opens the possibilty for future work, analyzing the same technique using the tecnologies WiMAX and the SIP VoIP protocol.**

***Index Terms*— Side; Channel; attacks; Timing; systems; countermeasures.**

I. Introducción

El SCA se considera una clase de ataque físico, en el cual un adversario trata de explotar la filtración de información que el dispositivo emite en la ejecución del algoritmo criptográfico basado en: la medición de tiempo, el consumo de potencia o las radiaciones electromagnéticas [1]. El ataque TA mide el tiempo que se demora una unidad en realizar una operación. Por medio de la medición precisa de la cantidad de tiempo requerida para generar operaciones de clave secretas, el atacante puede encontrar exponentes de Diffie-Hellman, factores de una clave RSA, y quebrar otros sistemas criptográficos [2]. Las implementaciones de algoritmos criptográficos por lo general ejecutan cómputos en tiempos no constantes debido a las optimizaciones de rendimiento. Si estas operaciones incluyen parámetros secretos, dichas variaciones en el tiempo pueden filtrar información, y con el conocimiento adecuado del criptosistema se puede llevar a cabo un análisis estadístico que desencadena en la recuperación de estos parámetros secretos [3]. Además, un TA es esencialmente una forma de obtener información privada de algún usuario a través de la medición dependiente del tiempo que el usuario necesita para llevar a cabo operaciones criptográficas. Una vez que se dispone de información suficiente, el criptosistema podría ser roto [4].

El ataque Power Analysis Attack (PA), es un ataque enfocado en el análisis del consumo de energía o potencia de una unidad mientras realiza una operación de cifrado [1]. Un equipo criptográfico, por medio de su consumo de energía puede proveer información relacionada con las operaciones que se está realizando y los parámetros involucrados en las mismas [3]. Este ataque puede ser divido en dos clases que son Simple Power Analysis (SPA) y Differential Power Analysis (DPA). En los ataques SPA, el enfoque apunta a interpretar el consumo de energía del equipo y deducir información acerca de las operaciones ejecutadas. En el caso de los ataques DPA, tienen como objetivo tomar ventaja de la dependencia de datos en los patrones de consumo de energía [1].

El ataque Electromagnetic Attack (EMA) se enfoca en las radiaciones electromagnéticas generadas por los componentes de una computadora o equipos electrónicos como parte de su funcionamiento. Un atacante puede observar estas emisiones y entender e inferir una gran cantidad de información acerca de los cálculos y datos que están siendo ejecutados. Similar a los ataques de consumo de energía, los ataques EMA pueden dividirse en dos tipos que son Simple ElectroMagnetic Analysis (SEMA) y Differential ElectroMagnetic Analysis (DEMA) [3].

El presente estudio tiene como objetivo analizar las implementaciones de los SCA con base en la técnica de TA

Francisco Bolaños Burgos es investigador de la Universidad de Especialidades Espíritu Santo de Guayaquil, e-mail: fcobolanos@uees.edu.ec
Luis García Tenesaca es investigador de la Universidad de Especialidades Espíritu Santo de Guayaquil, e-mail: lfgarcia@uees.edu.ec
Antonio Cevallos Gamboa, es investigador de la Universidad de Especialidades Espíritu Santo de Guayaquil, e-mail: acevallos@uees.edu.ec

tomando los 22 estudios seleccionados por juicio de expertos. Con la finalidad de conocer si existen relaciones entre los modos de operación, activos de información atacados y las contramedidas. En el capítulo II se describen las principales características de los 22 artículos seleccionados. En el capítulo III se analizan los estudios previo en términos delos activos de información más atacados, las contramedidas y los modos de operación utilizados. Finalmente en el capítulo IV se concluye en función del objetivo planteado y se plantean trabajos futuros.

II. Estudios Representativos

La selección de los artículos científicos se dio mediante juicio de expertos. Se realizó un focus group de seis expertos. Dos de ellos son peritos en criptografía, uno en matemáticas y los dos restantes en algoritmia. Los expertos efectuaron dos rondas de selección para decidir la inclusión de los artículos en este estudio.

El origen del SCA se da en el año 1965, la agencia británica de inteligencia (MI5), trató de penetrar el algoritmo de cifrado usado por la embajada egipcia en Londres. Pero sus esfuerzos fueron obstaculizados por los límites de su poder computacional en aquel tiempo. Se colocó un micrófono cerca del rotor de la máquina de cifrado, usado por los egipcios, para espiar el sonido de los clics que la máquina realizaba. MI5 dedujo correctamente la posición central de 2 o 3 rotores de la máquina. Esta información adicional redujo el esfuerzo de cálculo que se necesitaba para romper el cifrado y MI5 pudo espiar la comunicación de la embajada durante años [3].

En [5] mencionó que, por medio de la medición cuidadosa de la cantidad de tiempo requerida para realizar operaciones de claves secretas, los atacantes pueden ser capaces de encontrar exponentes Diffie-Hellman, factores RSA y quebrar otros criptosistemas. El ataque propuesto en [6] presenta mejoras al ataque presentado en [5], realizando una implementación, en tarjetas inteligentes, capaz de romper una clave de 512 bits en pocos minutos. En [7] se introduce un nuevo tipo de TA que permite la factorización de un módulo RSA si la exponenciación matemática con el exponente secreto usa CRT y el algoritmo de MM. En [8] el ataque *Divide and Conquer* trató de recuperar pequeñas porciones de claves criptográficas, dividiendo las claves en pequeñas partes que permitan realizar búsquedas exhaustivas y puedan ser manejadas por separado. En [9] un nuevo ataque es expuesto por medio del análisis de las variaciones de tiempo en una implementación de multiplicación modular, si un algoritmo de exponenciación binaria es utilizado, sólo un pequeño número de observaciones es necesario para realizar un ataque exitoso. En [10] se presentó un ataque mejorado eficientemente por medio de la observación de los factores básicos hacia el uso racional de potentes herramientas estadísticas. En [11] se enseñó una mejora al ataque [9] usando métodos estocásticos adecuados, la eficiencia del ataque se mejora por un factor de 5 en tablas de 2 bits.

En [12] se exhibió un ataque que aplica la teoría de decisión estadística y optimizó dos variantes relacionadas a los TA introducidos en [5; 6]. En [13] se mostró un ataque que apunta a las implementaciones del algoritmo MM y adoptado la metodología descrita en [9], ha sido demostrado en algunos algoritmos de exponenciación estándar, como MM, se realizan restas condicionales produciendo suficiente información para deducir el valor del exponente secreto. En [14] se expuso que una vez que las claves fueran lo suficientemente largas, aumentar la longitud de las mismas puede tender a la disminución de su seguridad y aumento en la fuga de información por canales laterales. En [15] se señaló un TA en la librería *Open-SSL*, en el cual mediante experimentos se expuso que las claves pueden ser extraídas de un servidor web. En [16] se exteriorizó un ataque en implementaciones desprotegidas de *Open-SSL* mejorando la eficiencia por un factor mayor a 10 al ataque previo demostrado en [15], en este nuevo ataque se explotó el comportamiento de MM en la fase de inicialización incrementando el número de multiplicaciones que proveen información útil para revelar información secreta. En [17] se propuso un método general para optimizar la eficiencia de los SCA por medio de métodos estocásticos avanzados, específicamente se aplicó el cálculo de procesos estocásticos y teorías de decisiones estadísticas. En [18] una extensión de [7] se dio en implementaciones RSA con CRT y MM, el ataque requirió el tiempo de procesamiento durante la exponenciación. En [19] se estudió el potencial de realizar TA remotos, como en una red de área local LAN o en Internet, ambos ataques requieren múltiples mediciones de tiempo de un evento en un servidor remoto y el filtrado de dichas mediciones para eliminar el ruido que puede ser causado por la red o hosts finales.

En [20] un esquema de TA avanzado en algoritmos criptográficos fue exhibido, donde el atacante puede usar el algoritmo enseñado para romper un sistema criptográfico por medio de la reconstrucción de la clave secreta. En [21] la idea principal fue explotar los mecanismos de certificado de caché soportados en las infraestructuras SIP VoIP, este ataque puede ser usado para revelar efectivamente el historial de llamadas de un grupo de usuarios de VoIP. En [22] se expuso un TA avanzado en RSA con CRT con una política de corrección de error, por medio de la corrección de errores una clave RSA de 1024 bits puede ser recuperada. En [4] para incrementar la factibilidad de un ataque de medición de tiempo, los autores propusieron un esquema mejorado de un ataque en la implementación RSA con CRT, el algoritmo proporciona un mecanismo de detección de error y corrección de estrategia que puede detectar y corregir decisiones erróneas realizadas por los atacantes. En [23] se presentó un TA en contra del algoritmo RSA con CRT usado en la librería *Polar-SSL*, la implementación de este ataque hace uso de una contramedida clásica para evitar los dos previos ataques propuestos en [7,15]. En [24] se demostró que la suposición de cegamiento del exponente prevendría los TA, no es generalmente cierta, aunque reduce enormemente el ataque. En [25] se expone que exclusivamente cegar el exponente ha sido asumido como una prevención para los ataques de medición de tiempo, aunque reduce significativamente en impacto de los mismos, no es generalmente verdadera y la eficiencia de este ataque es mayor en comparación con [24].

III. Análisis de los Estudios

En el Apéndice se muestra de una manera holística y resumida todos los papers analizados. Exponiendo de manera cronológica los ataques ejecutados, los autores de los mismos, el país de origen de los estudios, los modos de operación empleados, los equipos atacados y las contramedidas desarrolladas por los autores.

Las columnas Autores y Años, demuestran la cantidad de ataques ejecutados a las implementaciones en RSA de acuerdo a los diferentes autores, también se evidencia el orden cronológico de los mismos. Con base en la cantidad de ataques por autor se observa que el autor con más ataques es Schindler, con un 45.5%. La cantidad de ataques realizados por año es notorio que el pico más alto es el 2005 con 18.2% de ataques ejecutados. La tendencia de los autores es desarrollar ataques más efectivos y precisos para poder recuperar claves RSA en tiempos menores y con la menor cantidad de mediciones de tiempo necesarios. El ataque desarrollado por Kocher [5], es un punto de partida para los diferentes ataques, los siguientes ataques tratan de mejorar los rendimientos de los ataques anteriores como Dhem y otros [6], trató de mejorar la efectividad del ataque Kocher. Otros ejemplos son los de: Schindler [11], Schindler y Walter [13], Chen, Wang y Tiang [22,4].

Adicionalmente, en la columna Modos de Operación, se observa los modos que los atacantes utilizaron con una paridad del 41% entre los que utilizaron sólo MM o MM con CRT. En los primeros TA realizados por Kocher [5], y Dhem y otros [6], se apuntaba a las implementaciones RSA que sólo utilizaban el algoritmo de multiplicación modular Montgomery. Esto era debido a la suposición que si estas implementaciones realizaban las reducciones con CRT eran impenetrables a este tipo de ataques. Sin embargo, Schindler [7], demuestra en su TA que dicha teoría era falsa, las implementaciones que usaban MM o MM con CRT son vulnerables a estos ataques de medición de tiempo. A partir de estas primicias los autores desarrollan diferentes tipos de ataques que apuntan a los diferentes modos de operación que utilizan las implementaciones RSA, en los cuales ambas son vulnerables a los TA.

En la columna Contramedidas, la principal defensa empleada por los autores es el cegamiento con 33%, introducida por Kocher [5], se utiliza como prevención para que los atacantes no puedan conocer los valores de entrada de las exponenciaciones modulares. La mayoría de los autores hacen uso del cegamiento del exponente como prevención de los TA. No obstante, Schindler [25], expone que sólo el cegamiento del exponente no es una defensa suficiente para estos ataques, aunque la eficiencia del ataque es reducida significativamente igual pueden ser vulnerados. Se recomienda que el cegamiento deba ser combinado tanto como la base y el exponente. Además, se observa con un 11% la segunda contramedida más empleada es la Reducción Adicional, pero también Schindler [25], recomienda que esta reducción debe ser evitada debido a que la fuga de información es mayor. Las contramedidas, se emplean dependiendo del ataque.

En la columna Equipos Atacados, se puede observar los diferentes tipos de equipos implementados. Siendo las primeras víctimas las tarjetas inteligentes con un 32% debido a que en estos equipos la facilidad de implementación de ataques era mayor como los demuestran Dhem y otros [6], Schindler [7,12], Schindler, Kouene y Quisquater [8], Walter [14], entre otros. Hasta el ataque presentado por Brumley y Boneh [15], existía la hipótesis que los servidores de uso general como los servidores web no podían ser víctimas. Sin embargo, esto fue desafiado y un TA fue ejecutado en servidores de código abierto *Open-SSL*. A partir de este ataque el enfoque a los servidores *Open-SSL*, con un 23%, fue implementado por varios autores como Aciiçmez, Schindler y Koç [16] o Chen, Wang y Tiang [22,4].

IV. Conclusiones y Trabajos Futuros

Tanto los sistemas computacionales como los de telecomunicaciones pueden ser víctimas de los TA, inclusive cuando estos sistemas emplean sistemas de cifrado, como RSA. Los resultados alcanzados demostraron que el enfoque de los autores es la creación de mejores ataques aprovechando la información de los anteriores y lo que en un principio se pensaba impenetrable se pudo demostrar que no es así. Se puede comprobar que la información relacionada a los sistemas de telecomunicaciones no es tan extensa como la presentada en los sistemas computacionales.

Una limitación al presente trabajo, es por ser un estudio exploratorio no hay acceso a toda la información y solo se pudo analizar 22 casos de los TA en RSA. También, al ser un tema nuevo en los sistemas de telecomunicaciones no hay suficiente información publicada en revistas científicas o de libre acceso. Además, no se poseen los equipos necesarios para realizar una simulación de una implementación de ataque en estos sistemas.

Por esta razón, como trabajo futuro se recomienda realizar investigaciones aplicando técnicas de meta análisis en estos sistemas para conocer otros tipos de defensas que puedan ser empleadas y también ataques adicionales. Además, enfocarse en otros tipos de SCA, en los sistemas de telecomunicaciones, para poder realizar una simulación de estos ataques en las tecnologías WiMAX, protocolo SIP de VoIP o en otros sistemas de telecomunicaciones. Por medio de estos nuevos ataques se podrá conocer si las contramedidas existentes son suficientes en los sistemas de telecomunicaciones. Adicionalmente se puede realizar una comparativa de los ataques estudiados en un mismo equipo.

Referencias

[1] F.-X. Standaert, «Introduction to Side-Channel Attacks,» de Secure Integrated Circuits and Systems, Boston, MA, USA, Springer US, 2010, pp. 27-42.

[2] R. Oppliger, Contemporary Cryptography (2nd Edition), Norwood, MA: Artech House, 2011.

[3] Y. Zhou y D. Feng, «Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing,» 2006. [En línea]. Available: https://www.eprint.iacr.org.

[4] C. Chen, T. Wang y J. Tian, «Improving timing attack on RSA-CRT via error detection and correction strategy,» Information Sciences, vol. 232, pp. 464-474, 2013.

[5] P. C. Kocher, «Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems.,» de Andvances in Cryptology - CRYPTO'96: 16th Annual International Cryptology Conference, 1996.

[6] J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestré, J.-J. Quisquater y J.-L. Willems, «A practical implementation of the timing attack,» de Smart Card Research and Applications: Third International Conference, CARDIS'98, 2000.

[7] W. Schindler, «A timing attack against RSA with the chinese remainder theorem,» de Cryptographic Hardware and Embedded Systems — CHES 2000: Second International Workshop, 2000.

[8] W. Schindler, F. Koeune y J.-J. Quisquater, «Improving divide and conquer attacks against cryptosystems by better error deteecction / correction strategies,» de Cryptography and Coding: 8th IMA International Conference, 2001.

[9] C. D. Walter y S. Thompson, «Distinguishing exponent digits by observing modular substractions,» de Topics in Cryptology - CT-RSA 2001: The Cryptographers' Track at RSA Conference 2001, 2001.

[10] W. Schindler, F. Koeune y J.-J. Quisquater, «Unleashing the full power of timing attack,» Catholic University of Louvain - Crypto Group, 2001, 2001.

[11] W. Schindler, «A combined timing and power attack,» Public Key Cryptography: 5th International Workshop on Practice and Theory in Public Key Cryptosystems, pp. 263-279, Febrero 2002a.

[12] W. Schindler, «Optimized timing attacks against public key cryptosystems,» Statistics & Risk Modeling, vol. 20, nº 1-4, pp. 191-210, 2002b.

[13] W. Schindler y C. D. Walter, «More detail for a combined timing and power attack against implementations of RSA,» Cryptography and Coding: 9th IMA International Conference, vol. 2898, pp. 245-263, 2003.

[14] C. D. Walter, «Longer keys may facilitate side channel attacks,» Selected Areas in Cryptography: 10th Annual International Workshop, SAC 2003, vol. 3006, pp. 42-57, 2004.

[15] D. Brumley y D. Boneh, «Remote timing attacks are practical,» Computer Networks, vol. 48, nº 5, pp. 701-716, 2005.

[16] O. Aciiçmez, W. Schindler y Ç. K. Koç, «Improving Brumley and Boneh timing attack on unprotected SSL implementations,» de Proceedings of the 12th ACM Conference on Computer and Communications Security, 2005.

[17] W. Schindler, «On the optimization of side-channel attacks by advanced stochastic methods,» Public Key Cryptography - PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, vol. 3386, pp. 85-103, 2005.

[18] Y. Tomoeda, H. Miyake, A. Shimbo y S. Kawamura, «An SPA-based extension of Schindler's timing attack against RSA using CRT,» de IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2005.

[19] S. A. Crosby, D. S. Wallach y R. H. Riedi, «Opportunities and limits of remote timing attacks,» ACM Transactions on Information and System Security (TISSEC), vol. 12, nº 3, 2009.

[20] R. Tóth, Z. Faigl, M. Szalay y S. Imre, «An advanced timing attack scheme on RSA,» de Telecommunications Network Strategy and Planning Symposium, 2008. Networks 2008. The 13th International, 2008.

[21] Z. Ge, F.-H. Simone, L. A. Martucci y S. Ehlert, «Revealing the calling history of SIP VoIP systems by timing attacks,» de 2009 International Conference on Availability, Reliability and Security, 2009.

[22] C. Chen, T. Wang y J. Tiang, «An improved timing attack with error detection on RSA-CRT,» 2010. [En línea]. Available: https://eprint.iacr.org/2010/054.

[23] C. Arnaud y P. Fouque, «Timing attack against protected RSA-CRT implementation used in PolarSSL,» de Topics in Cryptology – CT-RSA 2013: The Cryptographers' Track at the RSA Conference 2013 , 2013.

[24] W. Schindler, «Exponent blinding may not prevent timing attacks on RSA,» 2014. [En línea]. Available: https://www.eprint.iacr.org/.

[25] W. Schindler, «Exclusive exponent blinding may not suffice to prevent timing attacks on RSA,» de Cryptographic Hardware and Embedded Systems – CHES 2015: 17th International Workshop, 2015.

Francisco Bolaños Burgos es ingeniero en Computación y magíster en seguridad informática aplicada de la Escuela Superior Politécnica del Litoral (ESPOL) en Guayaquil, Ecuador. Se desempeña como director de la Maestría en Auditoría de Tecnologías de la Información (MATI). Enseña criptografía, hackeo ético y seguridad de la información en la Facultad de Postgrados en UEES. Sus líneas de investigación son: seguridad de la información y herramientas de evaluación (rubrics y scripts).

Luis García Tenesaca es ingeniero en Telecomunicaciones de la UEES. Se desempeña como Account Manager de Huawei Technologies CO., LTD. Sus líneas de investigación son: protocolos y estándares de comunicación.

Antonio Cevallos Gamboa es ingeniero en sistemas, magíster en Sistemas de Información Gerencial y Administración de Empresas, PhD candidate de la Universidad Del Rosario en Bogotá, Colombia. Es decano de la Facultad de Ingeniería en Sistemas Telecomunicaciones y Electrónica (FISTE). Enseña escritura académica, metodología de la investigación y sistemas de información en la Facultad de Postgrado en UEES. Sus líneas de investigación son: sistemas de información, Tics, innovación y liderazgo tecnológico.

APÉNDICE

Tabla 1: Resumen de los diferentes tipos de ataques ejecutados en implementaciones RSA.

#	Título	Autores	País	Año	Modos de Operación	Equipos Atacados	Contramedidas
1	Timing Attacks en implementaciones de Diffie-Hellman, RSA, DSS y otros sistemas	Kocher, P.	EEUU	1996	MM	Computadora	Enmascaramiento. Mediciones Imprecisas. Cegamiento.
2	Una implementación práctica de un Timing Attack	Dhem, J-F., Koeune, F., Leroux, P-A., Mestré, P., Quisquater, J-J., Willems, J.	Bélgica	2000	MM	Tarjeta Inteligente	Reducción Adicional. Cegamiento.
3	Un Timing Attack en contra de RSA con CRT	Schindler, W.	Alemania EEUU	2000	MM CRT	Tarjeta Inteligente	Reducción Adicional. Cegamiento.
4	Mejora de ataque Divide and Conquer en contra de criptosistemas por mejor detección de error / estrategias de corrección	Schindler, W., Koeune, F., Quisquater, J-J.	Alemania Bélgica	2001	MM y MM CRT	Tarjeta Inteligente	No especifica
5	Distinción de dígitos de exponentes por observación de sustracciones modulares	Walter, C.D., Thompson, S.	Inglaterra	2001	MM	Computadora	Modificación del Exponente.
6	Desencadenamiento de todo el poder de un Timing Attack	Schindler, W., Koeune, F., Quisquater, J-J.	Alemania Bélgica	2001	MM	Tarjeta Inteligente	No especifica
7	Un Timing and Power Attacks combinados	Schindler, W.	Alemania	2002a	MM	Computadora	Modificación del Exponente.
8	Timing Attacks optimizados en contra de sistemas criptográficos de clave pública	Schindler, W.	Alemania	2002b	MM	Tarjeta Inteligente	No especifica
9	Mayor detalle para un ataque combinado de Timing y Power Attacks contra implementaciones de RSA	Schindler, W., Walter, C.D.	Alemania Inglaterra	2003	MM	Computadora	Cegamiento. Modificación del Exponente.
10	Las claves largas pueden facilitar los Side Channel Attacks	Walter, C.D.	Inglaterra	2004	MM	Tarjeta Inteligente	Cegamiento. Reducción Adicional.
11	Los Timing Attacks Remotos son prácticos	Brumley, D., Boneh, D.	EEUU	2005	MM CRT	Open-SSL	Cegamiento. Descifrado Independiente. Descifrado Cuantificado.
12	Mejora al ataque de Brumley y Boneh en implementaciones desprotegidas de SSL	Aciiçmez, O., Schindler, W., Koç, Ç. K.	EEUU Alemania	2005	MM CRT	Open-SSL	Cegamiento.
13	Optimización de Side Channel Attacks por métodos estocásticos avanzados	Schindler, W.	Alemania	2005	MM y MM CRT	Tarjeta Inteligente	Cegamiento. Tiempos de procesamiento constantes.
14	Una extensión SPA del TA de Schindler en contra de RSA usando CRT	Tomoeda, Y., Miyake, H., Shimbo, A., Kawamura, S.	Japón	2005	MM CRT	No especifica	Reducción Adicional. Cegamiento.
15	Oportunidades y límites de Timing Attacks remotos	Crosby, S.A., Wallach, D.S., Riedi, R.H.	EEUU	2007	No especifica	Open-SSL	No especifica
16	Un esquema avanzado de Timing Attack en RSA	Tóth, R., Faigl, Z., Szalay, M. Imre, S.	Hungría	2008	MM	No especifica	No especifica
17	Desvelamiento del historial de llamadas de SIP en los sistemas de VoIP por medio de Timing Attacks	Zhang, G., Fischer-Huebner, S., Martucci, L. A., Ehlert, S.	Varios	2009	No especifica	SIP-VoIP	Evitar Certificado Cache. Tiempo de Proxy Uniforme.
18	Un Timing Attacks mejorado con detección de error en RSA-CRT	Chen, C.,Wang, T., Tiang, J.	China	2010	MM CRT	Open-SSL	Cegamiento.
19	Mejora de los Timing Attacks en RSA con CRT por medio de la detección de errores y la estrategia de corrección	Chen, C., Wang, T., Tiang, J.	China	2013	MM CRT	Open-SSL	Cegamiento.
20	Timing Attacks en contra de implementaciones protegidas de RSA con CRT usados en Polar-SSL	Arnaud, C., Fouque, P.-A.	Francia	2013	MM CRT	Polar-SSL	Cegamiento. Modificación del Módulo. Modificación generación de clave.
21	Cegamiento del exponente puede que no prevenga Timing Attacks en RSA	Schindler, W.	Alemania	2014	MM CRT	No especifica	Evitar Reducción Adicional. Cegamiento combinado.
22	Exclusivamente cegamiento del exponente puede no ser suficiente para prevenir Timing Attacks en RSA	Schindler, W.	Alemania	2015	MM CRT	No especifica	Evitar Reducción Adicional. Cegamiento combinado.

Revisión Sistemática de Literatura: Inyección SQL en Aplicaciones web
Systematic Literature Review: SQL Injection in Web Applications

Jesennia Iñiguez-Banegas, Rene Guaman-Quinche, Robert Figueroa-Diaz and Freddy Ajila-Zaquinaula

***Resumen*— La inyección SQL es una vulnerabilidad de seguridad que afecta a las aplicaciones web. Esto ocurre cuando se inserta una consulta SQL (código malicioso), por medio de las entradas de una interfaz de cliente permitiendo leer y modificar la información. El presente artículo detalla el proceso de la revisión sistemática de literatura sobre estudios primarios que plantean propuestas y solución acerca de inyección SQL. Se siguió el protocolo propuesto por Bárbara Kitchenham y se revisó un total de 9 estudios de varias revistas y conferencias. Las investigaciones sobre inyecciones SQL es todavía un tema abierto, se ha obtenido propuestas para la prevención y detección de la misma. Una de ellas es Hibrid Modeling Framework que hace frente a las vulnerabilidades de inyección SQL en la fase de diseño. Las soluciones expuestas son muchas y diversas, enfocadas en la prevención y detección de vulnerabilidades de inyección SQL.**

***Palabras clave*— mecanismos de seguridad, inyección SQL, frameworks de desarrollo, ataques inyección SQL, seguridad de aplicaciones web.**

***Abstract*— SQL injection is a security vulnerability that affects web applications. This occurs when a SQL (malicious code) query is inserted through the inputs of a client interface allowing you to read and modify information. This article details the process of systematic review of literature on primary studies that raise proposals and solution about SQL injection. Barbara Kitchenham proposed protocol was followed and a total of 9 studies of various journals and conferences was reviewed. Research on SQL injections is still an open issue, it has been obtained proposals for the prevention and detection of it. One is Hibrid Modeling Framework that addresses SQL injection vulnerabilities in the design phase. Exposed solutions are many and diverse, focused on prevention and detection of SQL injection vulnerabilities.**

***Index Terms*— security mechanisms, SQL injection, development frameworks, SQL injection attacks, web application security**.

Este artículo fue enviado para su revisión el 15 de Agosto de 2016

J. Iñiguez-Banegas egresada de la Carrera de Ingeniería en sistemas de la Universidad Nacional de Loja (e-mail:jesenniaib@gmail.com)

R. Guaman-Quinche y R. Figueroa-Diaz, Docentes Investigadores de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja (e-mail: rguaman, roberth.figueroa, pfordoñez@unl.edu.ec)

F. Ajila docente Investigador de la Escuela de Ingeniería Industrial, Facultad de Mecánica de la Escuela Superior Politécnica de Chimborazo (e-mail: freddy.ajila@espoch.edu.ec)

I. INTRODUCCIÓN

LA seguridad del software es una inquietud cada vez más significativa para las instituciones del sector público o privado. Sin embargo, pocos programadores abordan este carácter de calidad de forma estratégica [1].

Los arquitectos y desarrolladores continuamente ponen un énfasis mayor en satisfacer los requerimientos prácticos y funcionales, y la seguridad usualmente es aplicada como un "adicional" para arreglar una vulnerabilidad durante o después de que la aplicación ha sido desarrollada [2].

Desarrollar código encaminado a la seguridad es una tarea que pocos la realizan por ser compleja y [3], por ello, asiduamente se recurre a la adopción y uso de frameworks que se orientan en satisfacer distintas áreas de la seguridad como por ejemplo el control de acceso a los distintos sistemas, el cifrado de la información y la validación de entradas, entre las más importantes [4].

Para mejorar la seguridad en los framework en el diseño de arquitectura, se consideran tres enfoques:

- *Ninguna adopción*: cuando la seguridad no se considera para el diseño de la arquitectura, sino, soluciones adicionales para cubrir aspectos puntuales;
- *Adopción a medias:* se usan frameworks de seguridad luego del diseño inicial de la arquitectura; y
- *Adopción total:* considera la seguridad desde el inicio dentro del diseño de la arquitectura e influye en todo el proyecto [5] [6].

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, por lo tanto se depende de lo sólido y flexible del mismo a la vez lo cual es un problema para el programador si no lo sabe utilizar [7][8]. Siendo así que el 80% de problemas de seguridad de sistemas web se debe que los programadores no configuran los mecanismos de seguridad de frameworks [9].

Las vulnerabilidades de aplicaciones Web se han convertido, en los últimos años, en una gran amenaza para la seguridad de sistemas informáticos [10]. Esta situación se explica por el aumento de la complejidad de tecnologías de la Web [11], por la evolución frecuente de estas tecnologías, por los ciclos cortos de desarrollo de aplicaciones Web durante el cual las actividades de prueba y validación son limitados, y también, en algunos casos, por la falta de seguridad habilidades y cultura de los desarrolladores [12].

Según Owasp en el año 2014 el 98% de las aplicaciones web son vulnerables, lo que da como resultados un promedio de vulnerabilidades por aplicación del 20%. El servicio de almacenamiento en la nube FireHost reporta que el número de ataques de inyección de código SQL fue de cerca del 69% en el año 2012. Según un reporte, los servidores localizados en centros de datos alrededor de Europa y EUA registraron al menos medio millón de estos ataques en abril y junio de 2012; menos de 300 mil fueron registrados durante el primer trimestre [13] [14].

De las estadista descritas sobre los ataque de Inyección SQL, es necesario que se generen investigaciones para contribuir a que la información de los usuario tenga los principios de integridad, disponibilidad y confidencialidad, por ello, el propósito de este artículo es mostrar el resultado de la revisión sistemática de literatura, que fue orientada en estudios actuales en inyección SQL, la sección II presenta la metodología para desarrollar la revisión sistemática y extracción de información basada en [15][16]. En la sección III presenta los resultados obtenidos en tablas de estudios relevantes y en la sección IV se discute los principales hallazgos y la sección V se define las conclusions del presente artículo.

II. METODOLOGÍA

Basado en la metodología de revisiones sistemáticas de Bárbara Kitchenham se elaboró un esquema para la revisión, selección y extracción de información quedando de la siguiente manera:

a. Pregunta de investigación.
b. Palabras clave.
c. Método de revisión.
- Fuentes y estrategias de búsqueda
- Cadenas de búsqueda,
- Criterios de selección de estudios.
- Extracción de información.

d. Estudios incluidos y excluidos

Además se utiliza Mendeley, como gestor bibliográfico para almacenar y organizar los estudios y sus referencias.

A. Pregunta de investigación.

Se dirigió el alcance de este trabajo sobre artículos relacionados a mecanismos de seguridad aplicando frameworks de desarrollo. La pregunta de investigación planteada es:

¿Qué tipos de estudios primarios existen sobre mecanismos de seguridad para inyección SQL en frameworks de desarrollo?

B. Palabras clave.

Se realizó una revisión de literatura previa, que consistió en analizar algunos documentos relacionados al tema que facilitan identificar las palabras claves obtenidas de los títulos, resúmenes e introducción.

En la tabla 1 se detalla la lista de palabras obtenidas a través del Keywords.Equations

TABLA 1. REVISIÓN PRELIMINAR Y TÉRMINOS.

Cód.	Titulo	Palabras clave
R01	Towards SQL Injection Attacks Detection Mechanism Using Parse Tree	SQL injection attacks, parse tree, detection, web environments.
R02	Securing Web Applications from Injection and Logic Vulnerabilities: Approaches and Challenges	SQL injection, Cross-site scripting, Business logic vulnerabilities, Application logic vulnerabilities, Web application security, Injection flaws
R03	Effective detection of vulnerable and malicious browser extensions	Browser extensions, Web security, Malware, Hidden Markov Model, JavaScript
R04	Mitigating SQL Injection Attacks Via Hybrid Threat Modelling	SQL Injection Attacks, Software Security, SDLC, SSDL, Hybrid Threat Modeling, Attack Trees, Misuse Cases, State Machines
R05	Securing Web Applications from Injection and Logic Vulnerabilities: Approaches and Challenges	SQL injection, Cross-site scripting, Business logic vulnerabilities, Application logic vulnerabilities, Web application security, Injection flaws

Una vez obtenidas las palabras claves descritas en la tabla 1, se puede realizar la construcción de la cadena de búsqueda.

C. Método de revisión

1) Fuentes y estrategias de búsqueda

- SCOPUS Library: https://www.scopus.com
- SCIENCEDIRECT Library: http://www.sciencedirect.com
- IEEEXPLORE Library: http://ieeexplore.ieee.org/

2) Cadenas de búsqueda

A partir de la pregunta de investigación, se definieron palabras clave para las búsquedas: Security mechanisms, SQL injection, development frameworks, SQL injection attacks, web application security.

Para generar la cadena de búsqueda se utilizaron los operadores lógicos “OR” y “AND”, quedando:
(Security mechanisms and SQL injection or SQL injection attacks and development frameworks or web application security).

Criterios de inclusión.

Es necesario aclarar que se consideró los siguientes criterios de búsquedas:
- Considerar sólo publicaciones desde el año 2014 en adelante.

- Los resultados de la búsqueda solo sean en el área de Ciencias y Computación.
- Las producciones científicas sean estudios primarios (artículos de conferencia, artículos de revista).
- La búsqueda por su relevancia científica será en el idioma inglés.
- Los estudios deben tener información relevante a la pregunta de investigación.

Criterios de exclusión.

Los estudios que no han sido relevantes en este estudio se han excluido mediante los siguientes criterios:

- Publicaciones informales que no siguen una metodología científica.
- Todas las que no cumplan con los criterios de inclusión.

Las cadenas de búsquedas(C) utilizadas fueron las siguientes:

Biblioteca digital de SCOPUS Library:

C01: TITLE-ABS-KEY (security mechanisms AND sql injection OR sql injection attacks AND development frameworks OR web application security) AND (LIMIT-TO (PUBYEAR , 2016) OR LIMIT TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014)) AND (LIMIT-TO (SUBJAREA , "COMP"))

Biblioteca digital de SCIENCEDIRECT Library:

C02: ALL (Security mechanisms and SQL injection or SQL injection attacks and development frameworks or web application security) AND LIMIT-TO (yearnav, "2016, 2015, 2014") AND LIMIT-TO (cids, "271887","Computers & Security") AND LIMIT-TO (contenttype, "JL, BS","Journal")

Biblioteca digital de IEEEXPLORE Library:

C01: (Security mechanisms and SQL injection or SQL injection attacks and development frameworks or web application security)

3) Criterios de selección de estudios

Obtenidos los resultados de las búsquedas es conveniente describir el criterio a seguir para la selección de estudios primarios, considerando los siguientes:

- Presenten en el resumen, información actual de mecanismos de seguridad para inyección SQL en frameworks de desarrollo.
- Contener información relevante para la revisión en la introducción o conclusión.

4) Extracción de información

Los criterios de selección de estudios establecen la pauta de extracción de información relevante para este trabajo. Por cada artículo seleccionado, se sintetizará al menos uno de los siguientes elementos:

- Propuestas o modelos para prevenir inyecciones SQL
- Resultados
- Conclusiones relevantes.

D. Estudios incluidos y excluidos

El criterio utilizado para la selección de artículos fue que aportaran sobre la existencia de mecanismos de seguridad en los framework.

Las búsquedas realizadas generaron 24 artículos, de los cuales se registraron 5 coincidencias, es decir el número de artículos revisados fueron 19, de los cuales se seleccionaron 9 artículos de acuerdo al criterio ya mencionado.

TABLA 2. ARTÍCULOS INCLUIDOS Y EXCLUIDOS

	Artículos			
Base de Datos	Encontrados	Coincidencias	Revisados	Seleccionados
Scopus	10	2	8	6
ScienceDirect	13	3	10	2
IEEE	1	0	1	1
Total	24	5	19	9

III. RESULTADOS

Las siguientes tablas muestra la información relevante extraída de cada uno de los artículos seleccionados.

TABLA 3. RESULTADOS DEL ARTÍCULO SA01.

Mecanismos de seguridad	Se exhibe un esquema novedoso que transforma automáticamente las aplicaciones web, haciéndolas seguras contra ataques de inyección SQL. Esta técnica analiza dinámicamente el tamaño resultado de la consulta desarrollador destinados a cualquier entrada, y detecta los ataques de comparar esto contra el resultado de la consulta real.
Resultados	La evaluación empírica demostró que IDL (Injection Detector Libraries) consume más tiempo para la detección de los algoritmos existentes, ya que incluye muchos pasos que se consideran métodos sintácticos para dar resultados más precisos. Sin embargo este es un método que puede detectar con mayor precisión que SQLIA.
Conclusiones relevantes	Mediante el uso de una variable de entrada de sustitución y desinfección basada en el tamaño de la consulta, es posible detectar y prevenir las consultas SQL que incluyen vulnerabilidades de inyección.

TABLA 4. RESULTADOS DEL ARTÍCULO SA02.

Mecanismos de seguridad	Una nueva técnica o método para detectar SQLIA mediante el modelado de las consultas SQL como una gráfica de tokens y el uso de la medida de centralidad de los nodos para entrenar a una máquina de vectores soporte (SVM).
Resultados	Los resultados experimentales demuestran que esta técnica puede identificar con eficacia las consultas SQL maliciosas con sobrecarga de rendimiento insignificante.
Conclusiones relevantes	El sistema no requiere la construcción de un modelo de uso normal de las consultas, ni requiere el acceso al código fuente.

TABLA 5. RESULTADOS DEL ARTÍCULO SA03.

Mecanismos de seguridad	Propone una metodología de la vulnerabilidad y ataque de inyección para SQLi y XSS se puede aplicar a una variedad de configuraciones y tecnologías. Se basa en la idea de que podemos evaluar diferentes atributos de los mecanismos de seguridad de aplicaciones web existentes mediante la inyección de vulnerabilidades realistas en una aplicación web y atacar de forma automática.

Resultados	Los resultados muestran que la inyección de vulnerabilidades y ataques es de hecho una forma eficaz para evaluar los mecanismos de seguridad y para señalar no sólo sus debilidades, si no también formas para su mejora.
Conclusiones relevantes	Concluye que aproximadamente la mitad de las vulnerabilidades SQLi provienen de la explotación de los campos numéricos.

TABLA 6. RESULTADOS DEL ARTÍCULO SA04.

Mecanismos de seguridad	Un mecanismo de detección concreta basado en DSD (Dynamic Detección SQLIAs) se plantea para detectar SQLIAs mediante el uso de árbol de análisis sintáctico. La principal ventaja de la propuesta es que no requiere acceder al código fuente de las aplicaciones si no que es incorporado para entornos web existentes.
Resultados	Los resultados experimentales demostraron que el mecanismo tiene una mayor precisión (99.9%), menor tasa de falsos positivos (2%) y falsos negativos cuando se detecta SQLIAs. Por lo tanto es un eficiente mecanismo de detección SQLAS para entornos web.
Conclusiones relevantes	El mecanismo propuesto no requiere acceder al código fuente de las aplicaciones. Esto significa que DSD se puede aplicar directamente a aplicaciones web existentes. Por lo tanto es un eficiente mecanismo de detección de SQLIAs para entornos web.

TABLA 7. RESULTADOS DEL ARTÍCULO SA05.

Mecanismos de seguridad	Propone una amenaza Hybrid Modeling Framework, polilla, para hacer frente a vulnerabilidades de inyección SQL en la fase de diseño, una fase de desarrollo temprana del SDLC (ciclo vital del desarrollo/diseño de sistemas).
Resultados	Como resultado de los puntos de entrada ampliado, los investigadores han desplazado los engranajes de los enfoques reactivos prevalentes de SQLIAs la prevención de una estrategia de gestión de riesgos proactiva llamada de amenaza de modeling un ejercicio realizado en la fase de diseño del SDLC.
Conclusiones relevantes	La seguridad es un proceso continuo que debe ser integrado en las aplicaciones desarrolladas de solicitud a través de la liberación de mantenimiento.

TABLA 8. RESULTADOS DEL ARTÍCULO SA06.

Mecanismos de seguridad	Un enfoque basado en la técnica HMM (Modelo oculto de Markov) para detectar las extensiones del navegador vulnerable y malicioso, ampliando y complementando las técnicas existentes. Estas técnicas se centran principalmente en el análisis de flujo de información.
Resultados	Se implementa en una herramienta de prototipo y evaluó utilizando un número de 387 extensiones de Mozilla Firefox. Indican que el enfoque no sólo detecta extensiones vulnerables y maliciosos conocidas, sino que también identifica previamente no detectados, extensiones con una sobrecarga de rendimiento insignificante. La precisión de falso positivo 97,68%.
Conclusiones relevantes	El número de muestras utilizadas durante la evaluación es pequeño para apoyar la eficacia de HMM, nuestro enfoque se puede utilizar como una técnica complementaria a los enfoques existentes.

TABLA 9. RESULTADOS DEL ARTÍCULO SA07.

Mecanismos de seguridad	Una revisión de la literatura que resume el estado actual de la técnica para asegurar las aplicaciones web de los principales defectos tales como errores de inyección y lógicas. Aunque existen diferentes tipos de errores de inyección, el alcance se limita a la inyección de SQL (SQLI) y Cross-site scripting (XSS), ya que son calificados como los mejores entre la mayoría de las amenazas de los diferentes consorcios de seguridad.
Resultados	Se necesita más investigación en el área de los defectos de fijación en el código fuente de las aplicaciones. La mayoría de los artículos se centran en la detección de los defectos y la prevención de ataques contra las aplicaciones web.
Conclusiones relevantes	A pesar de que varios enfoques están disponibles para asegurar las aplicaciones web de SQLI y XSS, son todavía muy extendido debido a su impacto y la gravedad. Este articulo propocion una revisión integral de los recientes avances en la obtención de las vulnerabilidades de inyección y la lógica de negocio de las aplicaciones web, y señala los problemas no resueltos que deben abordarse.

TABLA 10. RESULTADOS DEL ARTÍCULO SA08.

Mecanismos de seguridad	Marco de seguridad adopción de Cloud Computing (CCAF) adecuada para negocios nubes. Se basa en el desarrollo y la integración de las tres principales tecnologías de seguridad: firewall, gestión de identidad y cifrado basado en el desarrollo de la empresa, sincronización de archivos y las tecnologías de Acciones.
Resultados	Los resultados en la primera hora y 24 horas pruebas mostraron que una CCAF protección de seguridad completa de múltiples capas podría bloquear y evitar la inyección de SQL para MySQL y MongoDB. En las pruebas de penetración, de seguridad en capas múltiples CCAF podría detectar y bloquear el 99,95%
Conclusiones relevantes	Una protección de seguridad multicapa completa CCAF podría bloquear toda inyección SQL y proporcionar una protección real a los datos.

TABLA 11. RESULTADOS DEL ARTÍCULO SA09.

Mecanismos de seguridad	Nueva metodología, basada en técnicas de agrupamiento de las páginas web, que está dirigido a identificar las vulnerabilidades de una aplicación web después de un análisis de cuadro negro de la aplicación de destino.
Resultados	Este enfoque también condujo al desarrollo de un nuevo escáner de vulnerabilidad denominada Wasapy.
Conclusiones relevantes	Enriquecer las gramáticas implementadas en Wasapy para permitir la generación de una variedad más grande de inyecciones que cubren las vulnerabilidades incluidos hasta el momento, así como las nuevas vulnerabilidades.

IV. DISCUSIÓN

Principales hallazgos

SA01: Se describe una técnica de adulteración positivo que caracterizan IZES el proceso de desinfección mediante el modelado de la forma en que una aplicación procesa los valores de entrada. En base al uso de una variable de entrada de sustitución y desinfección basada en el tamaño de la consulta, es posible detectar y prevenir las consultas SQL que incluyen vulnerabilidades de inyección. La evaluación empírica demostró que IDL (Injection Detector Libraries) a pesar de consumir más tiempo en la detección de algoritmos, es eficaz contra el conjunto de pruebas SQLIA. Los IDL tienen tres pasos principales: Paso 1 comprueba los patrones de ataque contra las expresiones regulares. Si ninguna regla coincide, entonces ese patrón de ataque se envía al sistema de detección. Paso 2 analiza el patrón de ataque utilizando nuestra base de datos interna, llamado un "conjunto de reglas," para clasificar la vulnerabilidad. Para evaluar la cadena de consulta, el IDL utiliza un analizador de SQL para dividirla en

una secuencia de símbolos que corresponden a palabras clave de SQL, operadores y literales. El IDL luego itera a través de los tokens y comprueba si las que no son literales contenen exclusivamente los datos de confianza. Si todas las fichas pasan esta comprobación, la consulta se considera segura y se puede ejecutar. Por último, el paso 3 se sutituye caracteres o cadenas vulnerables en la consulta SQL. En particular, esto incluye funciones que eliminan o sustituyen ciertos caracteres o cadenas de su entrada.

SA02: Muestra una nueva técnica para detectar SQLIA mediante el modelado de las consultas SQL como una gráfica de tokens y el uso de la medida de centralidad de los nodos para entrenar a una máquina de vectores soporte (SVM). El enfoque fue diseñado para trabajar en la capa de base de datos y servidor de seguridad se implementó en un prototipo llamado SQLiGoT, además utiliza las funciones disponibles en la mayoría de los lenguajes de programación modernos y puede ser portado a otras plataformas sin necesidad de grandes modificaciones. El sistema fue probado exhaustivamente el uso de grafos no dirigidos y dirigidas con dos diferentes métodos de borde de ponderación. Proponen diseños alternativos de la SVM clasificador, que consiste en simples y múltiples, probados y comparados. Los resultados experimentales obtenidos en cinco aplicaciones web totalmente vulnerables confirman la eficacia que tiene. El sistema no requiere la construcción de un modelo de uso normal de las consultas, ni requiere el acceso al código fuente.

SA03: La metodología propuesta ofrece un entorno práctico que se puede utilizar para probar mecanismos de contramedida (como los sistemas de detección de intrusos (IDS), escáneres de vulnerabilidades de aplicaciones web, paredes de fuego de aplicaciones web, analizadores de código estático, etc.), y el tren evaluar los equipos de seguridad, ayudar a estimar las medidas de seguridad (como el número de vulnerabilidades presentes en el código), entre otros. Esta evaluación de herramientas de seguridad puede realizarse en línea mediante la ejecución del inyector de ataque, mientras que la herramienta de seguridad también está en marcha; o fuera de línea mediante la inyección de un conjunto representativo de las vulnerabilidades que se pueden utilizar como banco de pruebas para evaluar una herramienta de seguridad. La metodología se llevó a cabo en una vulnerabilidad de hormigón y herramienta del inyector Ataque (Vait) para aplicaciones web. El primero en evaluar la eficacia de Vait en la generación de un gran número de vulnerabilidades realistas para la evaluación en línea de herramientas de seguridad, en particular los escáneres de vulnerabilidades de aplicaciones web. El segundo para mostrar cómo se puede explotar las vulnerabilidades inyectadas para lanzar ataques, permitiendo la línea evaluación de la eficacia de los mecanismos de contramedida instalados en el sistema de destino, en particular un sistema de detección de intrusos. Estos experimentos ilustran cómo propuesta se puede utilizar en la práctica, no sólo para descubrir debilidades existentes de las herramientas analizadas, sino también para ayudar a mejorarlas.

SA04: Propone un interesante mecanismo de detección concreta basada en DSD, se plantea para detectar SQLIAs mediante el uso de árbol de análisis sintáctico. La principal ventaja de la propuesta es que no requiere acceder al código fuente de las aplicaciones si no que es incorporado para entornos web existentes. El DDS consiste en cinco unidades: Collector1, Collector2, Repositorio1, Repositorio2, y Agente y SQLIAs. El mecanismo consta de dos fases: de clasifición y detección. Cuando un usuario envía una solicitud HTTP a una aplicación, la fase de clasificación está involucrado para identificar la solicitud si se trata de primer acceso en tiempo o tiempo de acceso no primero. Después de eso, la fase de detección proporciona una detección SQLIA para esta aplicación en los dos casos anteriores. La exactitud de este mecanismo es más de 99.9% para cada tipo de aplicación típica, la tasa de falsos positivos es menos de 2% para cada tipo de aplicaciones.

.

SA05: Estudio de amenaza Hybrid Modeling Framework, polilla, para hacer frente a vulnerabilidades de inyección SQL en la fase de diseño, una fase de desarrollo temprana del SDLC, el modelado de amenazas implica el descubrimiento de la superficie de ataque explotable del activo de software mediante el examen de todos los límites de confianza, el flujo de datos, incluyendo los caminos de entrada y salida de todos los puntos de entrada. Los casos de mal uso (CUG), árboles de ataque (ATS) y máquinas de estado de comportamiento (MAN) se combinan en una técnica híbrida para diseñar un modelo de amenazas necesaria para proporcionar los requisitos de seguridad optimas y activos de software, concluyendo que la seguridad es un proceso continuo que debe ser integrado en las aplicaciones desarrolladas de solicitud a través de la liberación de mantenimiento.

SA06: Se enfoca en técnica basada en HMM (Modelo oculto de Markov) para detectar las extensiones del navegador vulnerable y malicioso, ampliando y complementando las técnicas existentes. Estas técnicas se centran principalmente en el análisis de flujo de información, para capturar los flujos de datos sospechosos, imponer la restricción de privilegios de llamadas a la API de extensiones maliciosos, aplicar firmas digitales para supervisar las actividades del proceso y el nivel de memoria, y permitir a los usuarios del navegador especificar las políticas con el fin de restringir las operaciones de extensiones. Se implementa en una herramienta de prototipo y evaluó utilizando un número de 387 extensiones de Mozilla Firefox. Indican que el enfoque no sólo detecta extensiones vulnerables y maliciosos conocidas, sino que también identifica previamente no detectados, extensiones con una sobrecarga de rendimiento insignificante. La precisión de falso positivo 97,68%.

SA07: Presenta una revisión sistemática de la literatura de los recientes avances en la obtención de las vulnerabilidades de inyección de las aplicaciones web. El objetivo de este estudio es resumir el estado actual de la técnica para asegurar las aplicaciones web de los principales defectos tales como errores de inyección y lógicas. Se exploran principalmente los siguientes puntos:

- Se analizan diversos tipos de vulnerabilidades y ataques que explotan estas vulnerabilidades en aplicaciones web.

- Se analizan los pros y los contras de los enfoques de mitigación para proteger las aplicaciones web de inyección y de negocios vulnerabilidades lógicas
- Proporciona información sobre las capacidades de los escáneres de vulnerabilidad existentes.
- Se destacan las aplicaciones web de código abierto que pueden utilizarse para la prueba y evaluación.

A pesar de que varios enfoques están disponibles para asegurar las aplicaciones web de SQLI y XSS, son todavía muy extendido debido a su impacto y la gravedad

SA08: framework de seguridad de varios niveles adopción de Cloud Computing (CCAF) adecuada para negocios en las nubes. Se basa en el desarrollo y la integración de las tres principales tecnologías de seguridad: firewall, gestión de identidad y cifrado basado en el desarrollo de la empresa, sincronización de archivos y las tecnologías de Acciones. Describe la tecnología de seguridad básica de Empresa sincronización de archivos y Compartir, la arquitectura y componentes en capas, y las tecnologías y resultados básicos de varias capas de experimentos a gran escala para las pruebas de penetración, inyección SQL y escaneo de datos. Dando como resultado en las pruebas de penetración que podría detectar y bloquear el 99,95% los virus troyanos y mantener un 85%, por encima de bloquear durante 100 horas continúas de ataques. Una protección de seguridad multicapa completa CCAF podría bloquear toda inyección SQL que proporciona una protección real a los datos. CCAF seguridad multicapa tenía tasa de 100% de no informar de falsa alarma.

SA09: nueva metodología que permite identificar automáticamente las vulnerabilidades residuales de una aplicación web a partir del análisis de la aplicación específica, siguiendo un enfoque cuadro negro puesto que no requiere detalles de la implementación del código fuente de la página Web. Está diseñado para poner de relieve los posibles escenarios de ataque, incluyendo la explotación de vulnerabilidades varios sucesivos que no son necesariamente independientes. Utilizan una herramienta Wasapy (Evaluación de la Seguridad Web de aplicaciones en Python.) de software utilizando el lenguaje Python, lo que facilita enormemente el manejo de conceptos HTTP (cookies, configuraciones, etc.). Se implementa un escáner de vulnerabilidades Web que contribuyen a enriquecer las gramáticas implementadas en Wasapy para permitir la generación de una variedad más grande de inyecciones que cubren las vulnerabilidades incluidos hasta el momento, así como las nuevas vulnerabilidades.

No hay una solución única hasta el momento que pueda eliminar las vulnerabilidades y prevenir ataques SQL. Por lo tanto, una serie de técnicas de mitigación debe ser empleado para frenar la propagación de los ataques y eliminar las vulnerabilidades SQL.

La solución más ideal es eliminar vulnerabilidades SQL de la raíz, es decir, el código fuente. Sin embargo, en las aplicaciones web en el mundo real, la obtención del código fuente o parches puede ser difícil. Por lo tanto, las técnicas de análisis estático son más útiles durante el desarrollo de la aplicación y antes del despliegue. Las técnicas de análisis dinámicos como técnica de pruebas de penetración pueden ser utilizadas para explotar las aplicaciones web durante el tiempo de ejecución con el fin de determinar si todavía son vulnerables a ataques SQL.

V. CONCLUSIONES Y TRABAJOS FUTUROS

El trabajo se centra en 9 artículos relacionados con la investigación de inyección SQL. Se identifica las soluciones, métodos, técnicas propuestas en los estudios. Las soluciones propuestas son muchas y diversas, en su mayoría se enfocan en la prevención de ataques de inyección SQL y detección de vulnerabilidades. Solo el estudio SA05, discute la eliminación de vulnerabilidades de inyección SQL a partir del código fuente, lo que es importante para prevenir ataques y ahorrar recursos en post-implementación. Los artículos SA01, SA02, SA08, hacen un análisis sintáctico a través de tokens, SVM (máquina de vectores soporte), IDL (Injection Detector Libraries) para la detección de SQLIA. Los documentos SA03, SA04, SA06, SA09, realizan análisis semánticos de diferentes mecanismos de seguridad como DSD (Detección SQLIAs Dinámico), IDS (Sistema de detección de intrusos), HMM (Modelo oculto de Markov), CCAF (Framework de adopción de Cloud Computing) y Wasapy en la detección de vulnerabilidades. Y un estudio proporciona una revisión integral de los recientes avances en la obtención de las vulnerabilidades de inyección. Todos los estudios hacen frente a los problemas relacionados con inyecciones SQL para eliminarlos. Pero los ataques son cada vez más frecuentes, así que la seguridad debe ser tratada en todas los faces de desarrollo considerando la seguridad desde el principio y en todo el ciclo de vida de la aplicación y usar framework para soportarla puede dar excelentes resultados. El análisis de estudios primarios indican la facilidad con la que puede ser vulnerada una aplicación cunado no se le asigna una prioridad adecuada a los controles de seguridad en las distintas etapas de desarrollo.

En el futuro, realizar una nueva Revisión Sistemática en cuanto a los de mecanismos de seguridad para mitigar la inyección SQL en aplicaciones web, compáralo con nuestro estudio y saber cuáles han sido los avances en la seguridad a partir de la publicación de mismo. Realizar estudios secundarios para obtener información acerca de que estudios existen de los diferentes tipos de escáneres de vulnerabilidades existentes para mitigar la Inyección SQL en las aplicaciones web y las diferentes vulnerabilidades de las aplicaciones web.

VI. ARTÍCULOS SELECCIONADOS EN LA REVISIÓN SISTEMÁTICA.

SA01.-Y. S. Jang and J. Y. Choi, “Detecting SQL injection attacks using query result size,” Comput. Secur., vol. 44, pp. 104–118, 2014.

SA02.-D. Kar, S. Panigrahi, and S. Sundararajan,

"SQLiGoT: Detecting SQL Injection Attacks using Graph of Tokens and SVM," Comput. Secur., 2016.

SA03.-J. Fonseca, N. Seixas, M. Vieira, and H. Madeira, "Analysis of field data on web security vulnerabilities," IEEE Trans. Dependable Secur. Comput., vol. 11, no. 2, pp. 89–100, 2014.

SA04.-T. N. Aung and S. S. Khaing, "Genetic and Evolutionary Computing," Adv. Intell. Syst. Comput., vol. 388, pp. 405–411, 2016.

SA05.-H. Omotunde and R. Ibrahim, "Mitigating SQL injection attacks via hybrid threat modelling," 2015 IEEE 2nd Int. Conf. InformationScience Secur. ICISS 2015, pp. 15–18, 2016.

SA06.-H. Shahriar, K. Weldemariam, M. Zulkernine, and T. Lutellier, "Effective detection of vulnerable and malicious browser extensions," Comput. Secur., vol. 47, pp. 66–84, 2014.

SA07.-G. Deepa and P. S. Thilagam, "Securing web applications from injection and logic vulnerabilities: Approaches and challenges," Inf. Softw. Technol., vol. 74, pp. 160–180, 2016.

SA08.-V. Chang, Y. H. Kuo, and M. Ramachandran, "Cloud computing adoption framework: A security framework for business clouds," Futur. Gener. Comput. Syst., vol. 57, pp. 24–41, 2016.

SA09.-R. Akrout, E. Alata, M. Kaaniche, and V. Nicomette, "An automated black box approach for web vulnerability identification and attack scenario generation," J. Brazilian Comput. Soc., vol. 20, no. 1, p. 4, 2014.

REFERENCIAS

[1] R. A. Oliveira, N. Laranjeiro, and M. Vieira, "Assessing the security of web service frameworks against Denial of Service attacks," J. Syst. Softw., vol. 109, pp. 18–31, 2015.

[2] M. Castro-león, F. Boixader, M. Taboada, D. Rexachs, E. Universitària, and T. Cerdà, "Servicios y Seguridad , un enfoque basado en estrategias de ataque y defensa," pp. 39–48, 2015.

[3] D. CAMACHO, G. MARTINEZ, and D. BIANCHA, "Diseño De Framework Web Para El Desarrollo Dinamico De Aplicaciones," no. 44, pp. 178–183, 2010.

[4] M. D. P. Salas-Zárate, G. Alor-Hernández, R. Valencia-García, L. Rodríguez-Mazahua, A. Rodríguez-González, and J. L. López Cuadrado, "Analyzing best practices on Web development frameworks: The lift approach," Sci. Comput. Program., vol. 102, pp. 1–19, 2015.

[5] H. Cervantes, R. Kazman, and J. Ryoo, "Seguridad y uso de Frameworks _ SG." p. SG # 47, 2015.

[6] A. R. Sartorio, G. L. Rodríguez, and M. Vaquero, "Investigación en el diseño y desarrollo para el enriquecimiento de un framework colaborativo web sensible al contexto," XIII Work. Investig. en Ciencias la Comput., pp. 1–5, 2011.

[7] C. García, R. Hervás, and P. D. A.-/9 L. B.-G. Gervás, "Una Arquitectura Software para el Desarrollo de Aplicaciones de Generación de Lenguaje Natural," Soc. Española para el Proces. del Leng. Nat. Proces. Leng. Nat., vol. 33, pp. 111–118 ST – Una Arquitectura Software para el De, 2004.

[8] G. Martínez Villalobos, G. D. Camacho Sánchez, and D. A. Biancha Gutiérrez, "Diseño de Framework web para el desarrollo dinámico de aplicaciones," Sci. Tech., vol. XVI, no. 44, pp. 178–183, 2010.

[9] H. T. Quinche, René Guamán, "Seguridad en Entornos Web para Sistemas de Gestión Académica," pp. 1–47.

[10] R. Akrout, E. Alata, M. Kaaniche, and V. Nicomette, "An automated black box approach for web vulnerability identification and attack scenario generation," J. Brazilian Comput. Soc., vol. 20, no. 1, p. 4, 2014.

[11] A. María Reina Quintero, "Separación avanzada de conceptos en entornos WEB.," pp. 3–16.

[12] G. Deepa and P. S. Thilagam, "Securing web applications from injection and logic vulnerabilities: Approaches and challenges," Inf. Softw. Technol., vol. 74, pp. 160–180, 2016.

[13] Owasp, "OWASP Top 10 - 2013," OWASP Top 10, p. 22, 2013.

[14] J. I. Calderón, "Seguridad en Aplicaciones Web."

[15] S. E. Group and R. Unido, "Directrices para la realización sistemática de la literatura críticas en Ingeniería de Software Sección de Control de Documentos," 2007.

[16] B. Kitchenham, "Procedures for performing systematic reviews," Keele, UK, Keele Univ., vol. 33, no. TR/SE-0401, p. 28, 2004.

Jesennia Iñiguez, Egresada de la Carrera de Ingieneria en Sistemas de la Univesidad Nacional de Loja. Líneas de interés seguridad web, redes y telecomunicacones. Docente en Unidad Educativa Calazanz (Septiembre 2013 – Enero 2014). Instructora de computación en Compucenter Technology (Sept. 2014 – Noviembre 2014). Técnico y operados de cibercafé en Gyg@net (Enero 2012 – Noviembre 2014). Ciudad Loja, Ecuador, 2016

Rene Guamán Quinché, Docente Investigador de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja. Líneas de interés en tecnologías web y móviles, sistemas distribuidos y paralelos. Máster en Sistemas Informáticos Avanzados en la Universidad del País Vasco. Ciudad Loja, Ecuador, 2016

Roberth Figueroa Díaz, Ingeniero en Sistemas Informáticos y Computación, graduado en la Universidad Técnica Particular de Loja - Ecuador, máster en Ingeniería del Software para la Web por la Universidad de Alcalá - España. Es certificado en Sistemas Linux otorgado por IBM Advanced Career Education y por Microsoft Certified Program. Docente en la Universidad Técnica Particular de Loja, Universidad Internacional del Ecuador y Universidad Nacional de Loja, así como analista desarrollador de software en la Unidad de Proyectos y Sistema Informáticos de la UTPL. Líneas de interés están la Ingenieria del Software, Ciencia de datos. Inteligencia Artificial, Internet de las cosas y Big Data.

Freddy Ajila, Docente Investigador de la Escuela de Ingeniería Industrial, Facultad de Mecánica de la Escuela Superior Politécnica de Chimborazo ESPOCH (Desde Octubre 2014 hasta la actualidad). Profesor de Sistemas Operativos, Arquitectura de Computadoras y Estructuras de Datos de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja – Ecuador (Abril 2013 – Julio 2014). Magister en Telemática graduado en la Universidad de Cuenca – Ecuador (Agosto 2011). Ingeniero en Informática graduado en la Universidad Técnica Particular de Loja – Ecuador (Junio del 2006). Activista de software libre, Administrador de servidores, redes y telecomunicaciones. Provincia de Chimborazo, Ciudad Riobamba, Ecuador, 2016.

Published by:

Escuela Politécnica Nacional
Facultad de Ingeniería de Sistemas
Ecuador

http://lajc.epn.edu.ec/
lajc@epn.edu.ec

November 2016