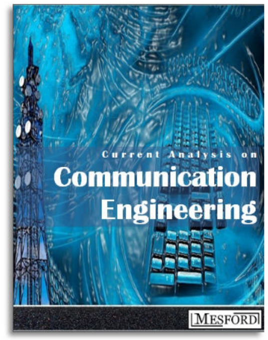

Autonomic Reflective Middleware for the Management of NANOdevices in a Smart Environment (ARMNANO)

Alberto López-Pacheco[1, 2] and Jose Aguilar[1,2,*]

[1]*CEMISID, Universidad de Los Andes, Mérida, Venezuela*
[2]*Tepuy R+D Group. Artificial Intelligence Software Development. Mérida, Venezuela*

Abstract:

This paper describes ARMNANO, which is a reflective middleware for the nanodevices management in smart environments. ARMNANO defines a set of mechanisms and services to deploy nanosensors and nanoactuators functionalities in a smart environment. The middleware is organized in layers, where each one is responsible for specific tasks. The base level comprises the nanodevices and the microgateway unit, meanwhile the upper layers contain the different services provided by the platform. Some of the layers are the Communication Management Layer, which is responsible for the nanocommunication, providing the specific communication protocols for each context; the Ontological Emergence Level, which is in charge of the management of the knowledge; or the Data Analysis Smart System, which is responsible for generating knowledge from the data acquired from the nanodevices, using the concept of "Autonomic Cycle of Data Analysis Tasks". Thus, nanodevices can act diagnosing in situ and transmitting through the nanoinfrastructure, to autonomously adapt the context. In this work, we detail the components of ARMNANO middleware, as well as there are 2 case studies in different contexts.

Publication History: Received: 11 December 2018 | Revised: 03 March 2019 | Accepted: 03 March 2019

Keywords:
Autonomic Computing, Reflexive Middleware, Smart Environment, Nanotechnology, Data Analysis, Nanosensors, Nanoactuators.

1. INTRODUCTION

Nanotechnology is relevant in the Ambient Intelligence (AmI). The nanotechnology has features such as real time monitoring, and notable sensitivity, flexibility and robustness. Such these characteristics introduce a real revolution at dissimilar areas of daily life, like health, military, transportation, communication, among others [1] [2]. For instance, several studies report that self-learning technology involving nanodevices have generated outstanding applications of AmI adapted to people needs [3].

Nanodevices provide advantages to the AmI architectures due to at a physical level they are connected to act in parallel. This association determines that an AmI can function based on the interconnections of embedded objects, allowing a faster task execution on a daily basis. Particularly, the nanostructure interconnection opens a new mechanism for the actuation in an AmI, in order to respond to different contexts [4].

On the other hand, nanosensors have fulfilled the gap in providing a full system characterization, which is not size dependent anymore, due to nanodevices capabilities and properties. The size influences the intrusiveness level, by allowing nanosensors to watch at not accessible places. Nevertheless, the sensors possess limitations to detect a rapid changing, giving rise to a possible false positive [5]. For instance, at the determination of glucose in blood, the concentration of glucose can vary faster than the sensor measurement frequency. For diabetic patients, this bias creates concerns, since the medical treatment change as a function of disease level, blood sugar level, gender, age, etc.

This constraint is avoided by an in situ diagnosis and fast autonomous reporting. In this sense, ARMNANO offers solutions in these contexts, due to it affords high responsiveness, autonomy, and interconnection of every important aspect in an AmI [6]. ARMNANO middleware follows a layer-by-layer approach, which can be used in different contexts that like taking advantage of the nanodevices capabilities. The agents of each layer act in the AmI for the data authentication, analysis of the nanodata, etc.

This paper is organized as follows: the section 2 presents a review of the previous works, then, at section 3 are defined the main features of the middleware, describing their layers and agents characterization. Also, in this section are outlined the

*Address correspondence to this author at the Universidad Autónoma de Madrid, Madrid 28049, Spain; E-mail: bazil.taha@uam.es

Mesford Publisher Inc
Office Address: Suite 2205, 350 Webb Drive, Mississauga, ON L5B3W4, Canada; T: +1 (647) 7109849 | E: cace@mesford.ca, contact@mesford.ca, https://mesford.ca/journals/cace/

ARMNANO main components, and a preliminary ontology of the middleware. At the section 4 are shown 2 case studies, and finally, the last section presents the conclusions of the ARMNANO middleware capabilities.

2. RELATED WORKS

The advent of smart systems involves specialized agents ranging either in the macro, micro or nanosize, which performs functional tasks related to sensing and executing in situ. These agents are known as nanosensors or nanoactuators. For that, it is required a new generation of middleware. In this section, we present some works ranging this domain:

Liu et al. [7] propose a parallel sensor middleware, called Impala. It possesses 2 main layers in its architecture. The upper layer is devoted to the protocols and programs, in order to gather all the context environment information. The lower layer operates through agents: an Application Adapter, an Application Updater and an Event Filter. Due to these agents, Impala is modular, adaptive, and repairable

Agilla is a middleware that supports communities of agents connected to a wireless network [8]. This middleware allows local and remote operations. Agilla handles situations in which the local decisions decrease significantly the quantity of transmitted data. An enhanced proposal of Agilla middleware was proposed afterwards by Lingaraj K. et al. [9], which adds a remote level for the multicasting wireless sensors. This middleware, called Eagilla, deploys a multicasting communication joining mobile agents in the network that sense, regulate and organize the communication. It eases the connectivity of the platform to support interoperability and heterogeneity in Wireless Sensor Networks (WSN).

A service-oriented component-based middleware, called MiSense, was developed by Khedo et al. [10], which is in charge of supporting a distributed sensor network on a wireless fashion. Its multitasking approach focuses into energy saving, hence providing an ecological energy distribution in an interconnected mesh. The planning of the distributed network in MiSense, settles that sensor at the base level captures the information, then communicates it to TinyOS, and subsequently to the data treatment layers in charge of filtering, patterning and converting the data into useful knowledge. This is carried out by the communication layer, the resource management layer, and the layer that provides direct services to users. The Domain layer is located upon the services layer, and it is the interface through which the results are provided. At this layer, the interconnectivity occurs.

Networked embedded environments need to be optimized, in order to have a better resource administration and energy management, among other things. A middleware proposed by Costa P. et al. [11] considers these issues, spreading out an architecture based on networked sensors and actuators. This middleware possesses certain similarities with the MAPEK structure. At the base level, the components include the specific OS and a platform for the Kernel implementation. The executable layer is composed of applications, for instance, relative to a disaster management scenario.

Finally, Balasubramaniam et al. [12] state the internet of nanothings as a domain that possesses multiple advantages. One of them is due to the possibility to build up embedded systems that monitor at real time and report instantly to upper layers; shortening steps and increasing the rapidness of the system analysis and evaluation. Also, they propose to include a self-awareness mechanism at the microgateway, and data analysis to ease the data management. The authors emphasize that the middleware is optimized with the nanodevices insertion, in the sense of saving energy and decreasing response time.

Previous middleware are very specific: for nano-environment or AmI, based on autonomic computing, etc. Some of the problems detected in the previous works are: some of them do not consider nanodevices as key components in the environment; and some of the middleware either for instance Impala, Agilla or eAgilla, are not completely autonomic. ARMNANO covers these aspects in a layered approach that consider an adaptable, reflexive, cloud-connected, and nanodevices-based architecture.

3. ARMNANO DEFINITION

ARMNANO is a middleware capable of measuring, analyzing, actuating and learning at the level of nanodevices in an AmI, In this sense, ARMNANO is ubiquitous, non-invasive and real time responding. It is a reflective middleware that incorporates the nanotechnology as a key actor at the base level, with a set of mechanisms for its management in the context of the AmI, which are offered by ARMNANO as services (See Fig. 1). For the deployment of these services, our middleware exploits the cloud computing paradigm.

The reflective middleware is composed of hierarchical levels, starting at the level of nanodevices, and levels of knowledge generation. Our middleware has specific components, in order to allow the communication between nanodevices, and additionally, autonomic cycles for data processing that allows the capture of the data by the nanosensors, and command execution by the nanoactuators. This platform is characterized by the following concepts:

Autonomy: it is the property of the system for monitoring, understanding and acting. In our case, the autonomic cycles of data analysis tasks carry out this property.

Reflexive: It is the capability of the architecture to think, and to adapt to the context of operation of the AmI. In our case, the autonomic cycles determine the actions to be executed by the nanoactuators, after to analyze the information on the context, to adapt it.

Self-learning: it is the capability of the system to learn new knowledge. In our middleware, each autonomic cycle processes data in order to generate knowledge, meaning that the system is able to learn.

Real Time: despite the reflexive ability, the system reacts immediately, in order to decipher the situations and achieve the issue resolutions in real-time.

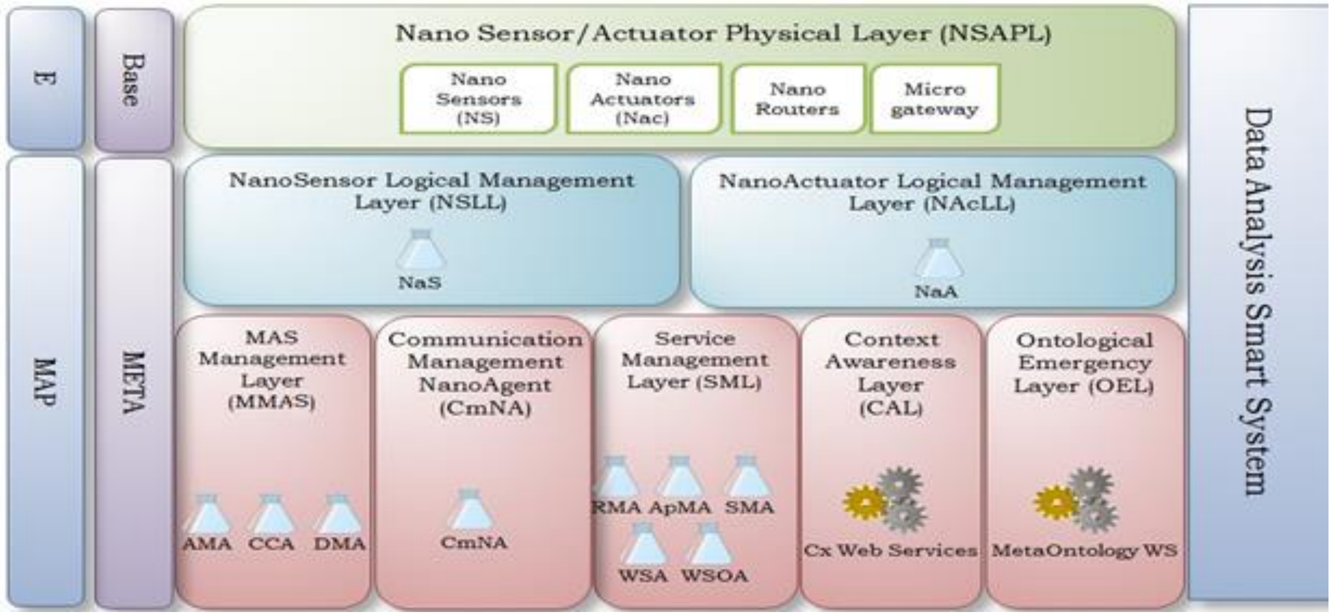

Fig. (1). ARMNANO Architecture.

As is shown in the Fig. (**1**), ARMNANO is a Reflexive Middleware divided into two main levels, the base level (where the Execution phase of the MAPE loop is carried out) and the meta level (where the Monitoring, Analysis and Planning phases of the MAPE loop are carried out).

Base Level: In this level is the nanodevices deployed in the AmI. The middleware guarantees a real time response, through the single tasks executed by the nanodevices. It is composed of Nanosensors and Nanoactuators that are primarily focused on monitoring the AmI and actuating according to the decisions made in the meta level. These nanostructures measure the variables of interest, as well as, perform the tasks that are commanded. The base level is called NanoSensor/Actuator Physical Layer (NSAPL), and its main components are:

Nanosensors; these nanodevices watch the variables of interest in situ (e.g. in the human body), in an ubiquitous fashion. One example of nanosensor is the Body Sensor Network (BSN), which is a network of sensors over the skin of the human body [13].

Nanoactuators; they are nanodevices that receive orders to act in the AmI. The nanoactuator are agents at the base level, that act according to the commands that are sent, normally by the DASS. Nanoactuators represent the first contact for task execution that the system will utilize.

Nanorouters; They represent devices that route the information between nanodevices in the AmI. They are located between the Nanosensor and Nanoactuator agents and the microgateway agent, for controlling the data routing. This nanodevice bridges the data transmission to the microgateway unit with less traffic.

Microgateway; it is devoted to route and authenticate the information in an appropriate fashion. This agent routes the information since the nanosensors nodes to the DASS, as well as, the data since the DASS to the nanoactuator agents.

Meta Level: it comprises the second level of the reflective middleware. At this level is done the introspection, by analyzing the message exchange in the upper layer, particularly between the nanosensors/actuators. Our middleware has 7 layers at this level. Some of these layers have been defined in previous works for the management of AmI [14] [17] [19], but others are specific due to the inclusion of the Nanotechnology in an AmI. The components defined in previous works are:

1. Multi-Agent Management Layer (MMAS): this layer is defined by the FIPA standard (see [14] for more details). It is composed of agents such as Agents Management Agents (AMA), Communication Control Agent (CCA), and Data Management Agent (DMA). These agents are defined in [14].

2. Service Management Layer (SML): this layer is transcendental for ARMNANO architecture, since it allows the connection between the MAS and SOA paradigms [15]. It means agents can consume web services in the cloud, and that the agent's tasks can be conceived as web services. This allows using the SaaS model of cloud computing in ARMNANO. The agents of this layer are: the Services Management Agent (SMA), the Web Service Agent (WSA), the Web Service Oriented Agent (WSOA), the Resource Management Agent (RMA), and the Application Management Agent (ApMA). These agents are defined in detail in [17] [19].

3. Context Awareness Layer (CAL): this layer offers context services through a cycle that is composed by discovering, modeling, reasoning and distribution of the context. This layer is based on the services offered in [18], where a context-aware middleware is proposed based on the cloud computing.

4. Ontological Emergence Layer (OEL): this layer offers ontological services based on [20]. Some of the services are:

automatic updating of ontologies to adapt them to the environment, and the Emergence of Ontologies.

The specific layers of ARMNANO are:

5. Nanosensor Logical Management Layer (NSLL): This layer characterizes each nanosensor as an agent. This layer saves metadata that describes the behavior of the agent. It possesses the NaS agent that characterizes each nanosensor as a whole and the applications running in it [18]. These agents are aware-context agents, and coordinate and cooperate with each other, in order to measure the variables of interest through a non-invasive fashion. The NaS receives data from the sensors individually, and then process them.

6. NanoActuator Logical Layer (NAcLL): this layer is where the nanoactuators are described. This layer will be operated by the NaA agents. Ontologically, nanoactuators execute orders, therefore, they do not possess elaborated algorithms, but to recognize the orders and trigger immediate actions. The commands are analyzed and sent from the DASS to the nanoactuators. The protocols are deployed here to control the nanoactuator agents action in situ.

7. Communication Management NanoAgent (CmNA): this layer is in charge of connecting nanodevices. The communication mechanisms can be Molecular Communication, Electromagnetic Communication, among others. The communication and data transmission are handled by the agents in it. This layer defines the communication protocols to receive and deliver data, from the nanosensors to DAAS, and from DAAS to the nanoactuators [20]. It ensures the system runs automatically to sense and act.

8. Data Analysis Smart System (DASS): it is a system that is transversal to the base and meta levels, which is in charge of processing data. Particularly, in it are defined the autonomic cycles of data analysis. An autonomous cycle is a closed loop of tasks of data analysis, which supervises permanently the process under study [28], [29]. It is a supervision cycle of processes based on data analysis tasks, that learn based on the experience. These autonomic cycles can execute data mining and semantic mining tasks, and can use different types of knowledge representations. In this way, this layer includes data treated mathematically to define patterns and trends, in order to delineate the actions that actuators must deploy at the base level.

The SML and MMAS layers provide operational services to the NSLL and NAcLL layers, and to the DASS. The MMAS and SML layers let the agents of ARMNANO: to know the services, agents and applications in the platform, to know the agents providing services and where they are, to know the ways to access these services, to expose the capabilities of the agents of ARMNANO and the web services in the cloud, among other things. These features are based on [14] [16] [17] [19], which support the SaaS model of cloud computing in ARMNANO.

The CAL and OEL layers work together to provide contextual services to the agents of the NSLL and NAcLL layers, and to the DASS, so that they can act intelligently and properly, to resolve the particular situations that arise in the nanostructured platform. Thus, these two layers provide the ubiquity property to the middleware, since the services are provided according to the context.

3.1. NanoAgents Interaction at the CmNA layer

The NSLL and NAcLL are connected at the physical level through the CmNA layer, as is shown in Fig. 2, using the microgateway.

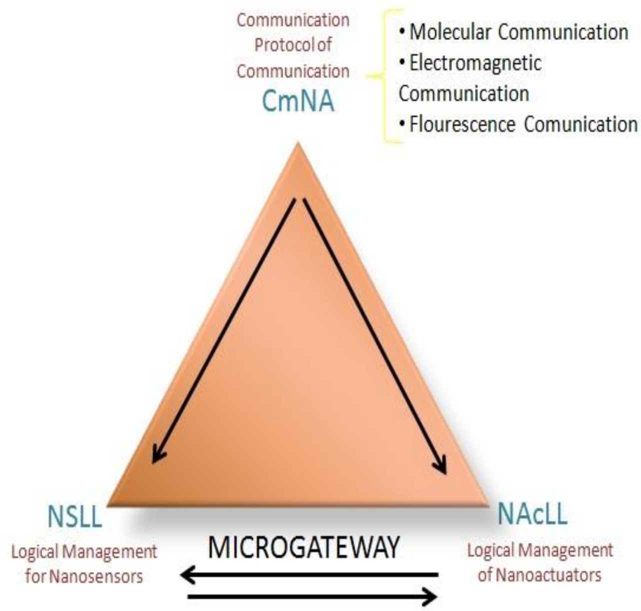

Fig. (2). Nanonetwork in ARMNANO.

Essentially, the microgateway will host the logical management level of the information. For example, the microgateway will host the data authentication to assure that the nanosensor nodes are measured accurately. On the other hand, CmNA designates the protocol for data transmission from the nanodevices to the microgateway.

The communication protocols handled at the CmNA (at the top of the Fig. **2**) will vary depending on several things: application, place of monitoring, and the monitored variable, among other things. For instance, to track the internal temperature of the body the nanosensor should be located in one specific place to use an electromagnetic communication protocol [27]. If the target is the white cell concentration, it should be used a Molecular Communication mechanism. In this sense, CmNA requires of evaluation and selection protocol mechanisms, that have to be dynamic and context-adaptable while acting in the AmI. Next works will describe in detail the architecture and components of the CmNA layer, and in particular, the behavior of the nanodevices and microgateway.

3.2. Data Analysis Smart Systems (DASS)

This is one of the main components of ARMNANO, in which the information is treated and stored to generate knowledge. Particularly, it uses the data captured by the nanosensors, in order to define the actions of the nanoactuators. Some of the knowledge generated are trends and patterns. For that, this component defines a set of autonomic cycles to analyze each situation individually in the AmI, such that each autonomic cycle has a specific goal to be reached.

Each autonomic cycle is defined by a set of data analysis tasks whose common objective is to achieve an improvement in the process under study. They interact with each other, and have different roles. In [28], [29], the autonomic cycles are defined as follows,

- Observe the system: this set of tasks must monitor the AmI and capture the data and information about the behavior of the different components of the AmI. Some of these data can be predicted, or can be estimated or extracted from different sources. In our case, the NaS agents execute these tasks.
- Analyze the system: this set of tasks has the goal to interpret, to understand, to diagnose, the current situation in the AmI. In our case, these tasks are executed at this level as services for the autonomic cycle.
- Make decisions to improve the AmI behavior: these tasks impact the dynamism of the AmI, because they make decisions about the processes. Specifically, these tasks define the command to be executed by the NaA agents [21].

In this way, the autonomic cycles allow the adaptation of the nanodevices for each situation, and a dynamic construction of knowledge models [22]. For that, the data analysis tasks use different techniques of data analysis, such as data mining, semantic mining or graph mining, for an efficient management and data exploitation.

In general, the autonomic cycles are based on the Autonomic Computing loop known as MAPEK [15][27]. In this loop, there are tasks that are in charge of monitoring, planning, analyzing and executing the plan, in order to respond to the different situations in the AmI. Each agent of ARMNANO has a role in the MAPEK loop. For example, the monitoring process is performed by the nanosensors agents of NSAPL. The NSLL, NAcLL and DASS perform the process of analysis and planning, in conjunction with the context-aware services and the meta-ontology. The execution of the plan is performed by the nanoactuators agents of NSAPL. In this sense, the meta level (Reflective Capacity) of the middleware occurs in the NSLL, NAcLL, CAL and OEL layers, while the base level (the system functionality) corresponds to the NSAPL layer.

The autonomic cycles provide several features that distinguish them of a classical middleware [23], some of which are:

- Self-configuration: the nanocomponents are automatically configured.
- Self-healing: it is capable of the correction and discovery of failures.

The main tasks of analysis and planning of the MAPEK loop are implemented in the DASS layer. Next works will describe in detail the architecture of the DASS layer, and in particular, its relationship with the cloud computing paradigm to provide the supervision process as a service.

3.3. A Preliminary Ontology for ARMNANO

The ontology based on nanodevices is required to clarify the key concepts handled at ARMNANO. The autonomic structure is based on definitions that characterize the data measuring, processing, storage and transmission, according to the MAPEK loop fitting ARMNANO [24]. The Table 1 describes a set of preliminary information classes managed by ARMNANO:

Table 1. Ontology Classes for ARMNANO

Classes	Subclasses
Communication Manager	*.Nanosensor Data*
	Logical Processing
	Information routing
Communication Protocol	*Electromagnetic (EC)*
	Molecular (MC)
	Fluorescence (FC)
	Radio Wave (RW)
	Infrared Light (IL)
Statistical Processing	*Data Authentication*
	Command Authentication
Mathematical Processing	*Charting*
	Patterning
	Predicting

Communication Manager, deals with the autonomic elements in charge of information processing.

- Nanosensor Data, it corresponds to the input information captured at the nanosensor level.
- Logical Processing, it defines the hardware at the microgateway, which manages the logical structure to treat data from nanosensors or nanoactuators.
- Information Routing, it is essentially an addresser of data. It guides the information to the less traffic hardware, allowing rapid processing.
- Communication Protocol, it defines the protocol for data transmission according to the resource capabilities.
- Electromagnetic Communication (EC), it defines data transmission in the terahertz band.
- Molecular Communication (MC), it is based on gradient changes of molecules on the in situ human body level. Variations of the molecular shift are monitored through gates positioned at certain positions in the organic fluid stream.
- Fluorescence Communication (FC), it occurs when there is an overlapping among a sender fluorophore

orbital and an acceptor fluorophore orbital. This transition can be encoded such as a binary language to transmit information describing the target.

- Radio Wave (RW), commercially known as Bluetooth. It refers to the short wavelength transmission in the ISM band from 2,4 to 2,485 GHz, typically employed in mobile devices.
- Infrared Light (IL), It deals with information transmitted in the form of long wave, at a spectrum section known as infrared zone.

Statistical Processing, this class treats the information coming from nanosensors and actuators based on statistical variables such as, average, standard deviation, variance, mode, and others [25][26].

- Data Authentication, data management performed at the microgateway to assure this is statically and mathematically approved.
- Command Authentication, statistic at the microgateway to certify the commands sent to the nanoactuator agents.

Mathematical Processing This class defines the mathematical platform used by ARMNANO.

- Charting, correspond to data plotting in order to analyze the behavior of the variables.
- Patterning, it studies the trending, in order to assign labels at the variable behavior.
- Predicting, it is defined by the functions that describe the future behavior of the variables. Thus, it is possible to extrapolate the variable trending.

4. CASE STUDIES

Below are described two case studies that show up a specific view and a broad view at the same health context. These two situations are complementary, and they comprise:

1. In situ target assessment, and

2. Smart room: target observation and actuation.

The main goal is to prove the ARMNANO processing capability, in the tasks of sensing, analysing, planning, self-learning, and executing, based on knowledge.

The variables in the AmI are time dependents. Thus, the middleware must assure a measurement frequency shorter than the change frequency of a variable, Vi. ARMNANO will self-adapt to the context based on the target requirements, so it looks to fit its performance features, such as: real time data analysis, and high interconnectivity.

In the following, the scenarios are analyzed. ARMNANO detects and predicts the evolution of the patient's condition at short and long term. The 1st scenario contemplates a patient that has a pain and inflammation in the hand due to unknown reasons, during 2 weeks. The 2nd scenario considers a permanent supervision in the smart room, with nanosensors capturing fixed variables indicating the individual's health conditions of the patient. The measured variables are blood pressure, temperature and/or the body heat distribution. The in situ nanosensors determine the close view, with specialized health monitoring/diagnosis, and the external nanosensors define a broad view, capturing fixed time dependent variables.

4.1. 1st Scenario

In this scenario, there is a set of tasks that are deployed in ARMNANO, to determine the patient's condition through the nanoagents. Therefore, there is a remarkable ARMNANO agents intrusiveness, as a consequence of the nanometric components. Our middleware carries out 4 tasks in this scenario, defining the Autonomic Cycle deployed in DASS (see Fig. **3**).

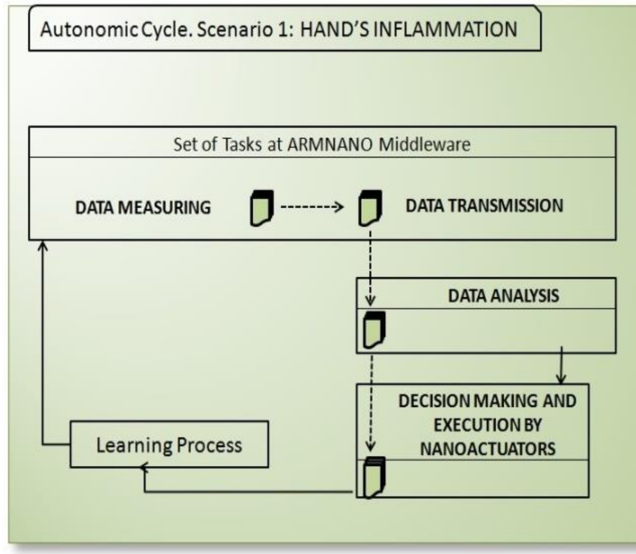

Fig (3). Autonomic Cycle of the first scenario.

The tasks of this Autonomic Cycle are:

A. Data Measuring, nanosensors are locally injected to monitor variables in situ, such as the coagulation and the inflammation level. Each one of these variables are monitored by different sensors. The monitored variables to this scenario change on every individual, due to every patient has a specific health condition. In this way, each individual despite being in the same smart room will be monitored differently.

B. Data Transmission, CmNA defines the protocol to transmit the information from nanosensors to nanoactuators, employing intermediate data transmitters, such as a nanorouters [12]. In particular, the measured data at the middleware is sent through the CmNA layer to the NaS agents, which carry out the logical data authentication. Also, the CmNA layer delivers the data to DASS (see Fig. 1).

C. Data Analysis, in the DASS are executed the data analysis tasks as services, supported by the agents of the MMAL, SML, CAL and OEL layers (see Fig. 2). All these agents act cooperatively, in order to analyze the data. In this sense, data analysis includes, but is not restricted to, pattern recognition, prediction, etc.

D. Decision Making and Executing, the decision made in DASS is transmitted until the nanoactuators and executed in situ. This information is recognized by the nanoactuators, in order to be executed.

4.2. 2nd Scenario

This scenario involves an external assessment of the patient, given by the nanosensors in the smart room. For this 2nd Scenario, it is deployed an autonomic cycle in DASS composed of 4 tasks (see Fig. **4**):

E. Data Measuring, in this case, the nanosensors measure fixed variables, in order to determine the patient's health condition from an external observation (not in situ, like in the first scenario). Nanosensors in the smart room unconditionally monitors temperature, blood pressure and body heat distribution to every individual in the room.

F. Data Transmission, measured data is sent to NSLL layer for authentication. In this case is used the communication services of ARMANO provided by the MMAL layer. This layer will deploy the required authentication process, such as statistics and recognition services. The validated data can now be sent to DASS.

G. Data Analysis, this step executes data analysis tasks to build mathematical models about the patient condition. Different to the 1st scenario , in this scenario, the CAL layer is not activated, due to the context is constant (see section 3). DASS uses the services of MMAL and SML (see Fig. 1), in order to offer the best diagnosis to the individuals.

H. Execution, in this scenario, the agents of the NAcL layer authenticate the decision before sending the command until the nanoactuators; otherwise, they carry out a new data analysis to achieve a more accurate decision. This action is sent to the NSAPL layer for its deployment.

In both scenarios, ARMNANO actuates essentially as a real time nanoagent-based middleware, which is omnipresent and iterate, up to achieve appropriate services to the targeted system. The autonomic cycles at both scenarios are complementary, so they are executed simultaneously and generate different data, such as the coagulation and inflammation level measured in situ (1st scenario) and the temperature, blood pressure and body heat distribution (2nd scenario). These data together provide an individual's diagnosis, and increase the robustness of the platform and its versatility for different health-related issues.

The autonomic cycles constitute a self-learning logical assembly of tasks that the system employs to act autonomously. The importance of ARMNANO relies in which it integrates different views to the same objective. ARMNANO is decentralized, learns and actuates autonomously, based on the layers distribution and their composing agents, stated in Fig. (**1**).

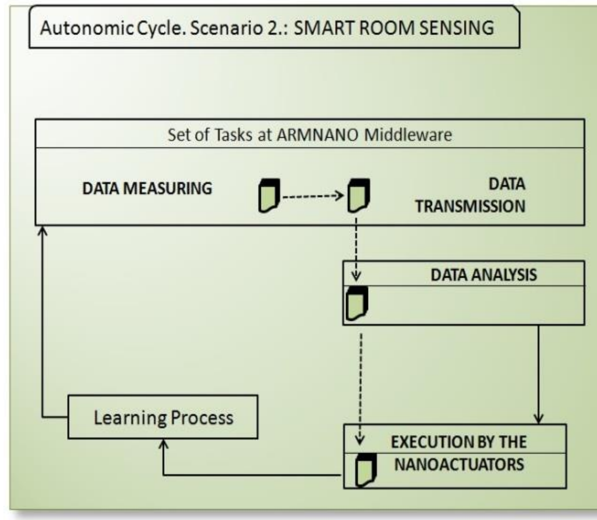

Fig. (4). Autonomic Cycle of the second scenario

5. CONCLUSION

This paper proposes a middleware based on the MAPEK loop, which involves nanodevices as physical agents. It allows performing real time monitoring, rapid response, autonomous analysis and high interconnection. The novelty of our architecture is that considers the deployment of nanoagents that act constantly to provide services in the AmI. To this end, it uses Autonomic Cycles as are shown in Fig. 3 and 4. The ARMNANO is flexible to integrate different views and create conclusions in the same objective, so it is capable of producing complementary knowledge. ARMNANO middleware is versatile, robust, and fast to autonomously work.

At the 1st scenario, the ARMNANO platform activates the autonomic cycle to determine the situation of the patient in situ, using a set of nanodevices for this purpose. The 2nd scenario offers more information that contributes to the individual's diagnosis, using external sensors in the smart room. This system is supported by a nanoactuator level, which allow for instance, the in-situ delivery of drugs, the destruction of strange body (e.x. tumors or fat), the cut nerves connection, etc.

Next works will focus on the development of the CmNA layer, which is one of the main layers of the middleware, particularly, in the implementation of the data authentication and communication protocols. Also, next works will be about the conception of the DASS layer and the Autonomic Cycles, and about the detailed definition of the ARMNANO ontology.

CONFLICT OF INTEREST

The author declares that there are no conflicts of interest.

REFERENCES

[1]. IAkyildiz I. Jornet J, The Internet of Nano Things,"IEEE Wireless Communications, 2010; 7(1): 58-63.

[2]. Dressler F. Fischer S., Connecting in-body nanocommunication wih body area networks: Challenges and Opportunities of the Internet of NanoThings, Nano Communication Networks, 2015; 6(2): 29-38.

[3]. D'Oro, S. Galluccio A. Morabito G. Palazzo S., A timing channel-Based MAC protocol for energy-efficient nanonetworks, Nano Communication Networks, 2015; 6(2): 39-50.

[4]. Alkyildiz I. Jornet J. Pierobon M., Nanonetworks: A New Frontier in Communications, Communications of the ACM, 2011; 54(11): 84-89.

[5]. Atakan B.. Akan O., Carbon Nanotube-based nanoscale Ad Hoc Networks,IEEE Communications Magazine, 2010; 48(6): 129-135.

[6]. Atakan B. Akan O., Body Area Nanonetworks with Molecular Communications in Nanomedicine, IEEE Communications Magazine, 2012; 50(1): 28-34.

[7]. Liu T. Martonosi M., Impala: A middleware system for managing Autonomic, Parallel Sensor Systems, Proceedings of the 9th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming; 2003 June 11-13; San Diego, USA: 107-118.

[8]. Fok C., Roman G. Lu C, Agilla: A mobile agent middleware for self-adaptive wireless sensor networks, ACM Transactions on Autonomous and Adaptive Systems, 2009; 4: 1-26.

[9]. Lingaraj K. Biradar V. Patil V, Eagilla: An Ehanced Mobile Agent Middleware for Wireless Sensor Networks, Alexandria Engineering Journal, 2018; 57(3): 1197-1204.

[10]. Khedo K. Subramanian R, A service-oriented component-based middleware architecture for wireless sensor networks, International Journal of Computer Science and Network Security, 2009; 9(3): 174-182.

[11]. Costa P. Coulson G. Gold G, The RUNES middleware for Networked Embedded Systems and Its Application in a Disaster Management Scenario, Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications, 2007 9-23 March; NY, USA: 1-10.

[12]. Balasubramaniam S. Kangasharju J, Realizing the Internet of Nanothings: Challenges, Solutions and Applications, IEEE Computer Society, 2013; 46(2): 62-68.

[13]. Zamani M. Tahoori M, Reliable Design for Crossbar Nanoarchitectures, Modelling and Optimization in Science and Technologies, 2017; 9: 421-444.

[14]. Aguilar J. Cerrada M., Mousalli G. Rivas F. Hidrobo F, A Multiagent Model for Intelligent Distributed Control Systems, Lecture Notes in Artificial Intelligence, Springer-Verlag, 2005; 368: 191-197.

[15]. Vizcarrondo J. Aguilar J. Exposito E. Subias A, ARMISCOM: Autonomic Reflective Middleware for Management Service Composition, Proceedings of the IEEE Global Information Infrastructure and Networking Simposium, 2012 4-6 December; Choroni, Venezuela: 1-8.

[16]. Sanchez M. Aguilar J. Cordero J. Valdiviezo P. Barba L. Chamba-Eras L, Cloud Computing in Smart Educational Environments: Applications in Learning Analytics as Service, In Advances in Information Systems and Technologies, In: Rocha Á., Correia A., Adeli H., Reis L., Mendonça Teixeira M. Eds. New Advances in Information Systems and Technologies. Springer, pp. 993-1002

[17]. Aguilar J. Valdiviezo P. Cordero J. Sánchez M., Conceptual Design of a Smart Classroom Based on Multiagent Systems, Proceeding of the Int. Conf. Artificial Intelligence; 2015 5-8 July, Las Vegas, USA: 471-477.

[18]. Aguilar J. Jerez M. Exposito E. Villemur T, CARMiClOC: Context Awareness Middleware in CLOud Computing, Proceedings of the Latin American Computing Conference 2015 9-23 Oct., Arequipa Peru.

[19]. Sanchez M. Aguilar J. Cordero J. Valdiviezo, Basic features of a Reflective Middleware for Intelligent Learning Environment in the Cloud (IECL), Proceeding of the Asia-Pacific Conference on Computer Aided System Engineering (APCASE), 2015; Quito Ecuador: 5-17.

[20]. Mendoca M. Aguilar J. Perozo N, MIR-EO: Middleware Reflexivo para la Emergencia Ontologica en Ambientes Inteligentes, Latin American Journal of Computing, 2016; 3: 16-21.

[21]. Portmann E. Finger M, Towards Cognitive Cities: Advances in Cognitive Computing and its Applications to the Governance of Large Urban Cities Springer, 2016.

[22]. Lindwer M. Marculescu D. Basten T, Ambient Intelligence Visions and Achievements. Linking Abstact Ideas to Real-World Concepts, Europe Conference in Design, Automation and Test, 2003 5-7 March; Munich, Germany.

[23]. Poslad F, Autonomous Systems and Artificial Life Ubiquitous Computing: Smart Devices, Environments and Interactions, Wiley, 2009.

[24]. Acampora G. Cook D. Rashidi P. Vasilakos A, A survey on Ambient Intelligence in Healthcare,Proceedings of the IEEE, 2013; 101(12): 2470-2494.

[25]. Balasubramaniam S. Jornet J. Pierobon M. Koucheriavy Y, Special Issue on the Internet of Nano Things, IEEE Internet of Things Journal, 2016; 3: 1-3.

[26]. Aguilar J. Jerez M. Mendonca M. Sánchez M, MiSCi: Autonomic Reflective Middleware for Smart Cities, Communications Computer and Information Science Series, 2016; 658: 241-253.

[27]. Baker N. Pappu R. Thomas D, Nanoparticle Ontology for Cancer Nanotechnology Research, Journal of Biomedical Informatics, 2011; 44: 59-74.

[28]. Aguilar J. Cordero J. Barba L. Sanchez M. Valdiviezo P. Chamba L, Learning Analytics Tasks as Services in Smart Classroom, Universal Access in the Information Society Journal, 2018; 17: 693–709.

[29]. Aguilar J. Cordero J. Buendia O, Specification of the Autonomic Cycles of Learning Analytic Tasks for a Smart Classroom, Journal of Educational Computing Research, 2017; 56(6): 866-891