# An Autonomic Cognitive Pattern for Smart IoT-based System Manageability: Application to Comorbidity Management

Emna Mezghani, Ernesto Expósito, Khalil Drira

# An Autonomic Cognitive Pattern for Smart IoT-based System Manageability: Application to Comorbidity Management

E.MEZGHANI, LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
Luxembourg Institute of Science and Technology, 5, Avenue des Hauts-Fourneaux, L-4362, Esch/Alzette, Luxembourg
E.EXPOSITO, Laboratoire LIUPPA, University of Pau and Adour Countries, Anglet, France
K.DRIRA, LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France

The adoption of the Internet of Things (IoT) drastically witnesses an increase in different domains, and contributes to the fast digitalization of the universe. Henceforth, next generation of IoT-based systems are set to become more complex to design and manage. Collecting real-time IoT generated data unleashes a new wave of opportunities for business to take more precise and accurate decisions at the right time. However, a set of challenges including the design complexity of IoT-based systems and the management of the ensuing heterogeneous big data as well as the system scalability; need to be addressed for the development of flexible smart IoT-based systems. Consequently, we proposed a set of design patterns that diminish the system design complexity through selecting the appropriate/combination of patterns based on the system requirements. These patterns identify four maturity levels for the design and development of smart IoT-based systems. In this paper, we are mainly dealing with the system design complexity to manage the context changeability at runtime. Thus, we delineate the autonomic cognitive management pattern, which is most mature level. Based on the autonomic computing, this pattern identifies a combination of management processes able to continuously detect and manage the context changes. These processes are coordinated based on cognitive mechanisms that allow the system perceiving and understanding the meaning of the received data to take business decisions, as well as to dynamically discover new processes meeting the requirements evolution at runtime. We demonstrated the use of the proposed pattern with a use case from the healthcare domain, more precisely the patient comorbidity management based on wearables.

Key Words: IoT-based system, Maturity Level, Autonomic Computing, Cognitive Computing, Design Patterns, Healthcare

## 1 INTRODUCTION

The emergence of the Internet of things (IoT) has revolutionized the healthcare through capturing real-time and individualized data concerning patients. It fosters providing patient centric management as well as preventing health complications. The wearable computing, as an example of IoT technologies, is fast gaining momentum. The use of wearable computing allows enhancing the patient's care and life style management through enabling remote data stream processing and detecting anomalies at the right time which profoundly impacts the decision process. Its utility goes further to enable disabled person's feeling motions and doing actions, despite their impairment [1].Real-time analyzing the data stemming from the patient wearable devices, and integrating them with the patient medical history and with medical knowledge could reach benefits for accelerating the decision-making and personalizing the patient treatment.

As these technologies sense the physical world, it is easy to collect data, but hard to manage. Automating the management of IoT-based systems may address the system complexity, accelerate and facilitate the interaction with the domain experts for better decision making. In software engineering, the autonomic computing initiative [2] has been proposed to enable the design of context-aware systems and automate their management based on the MAPE-K loop pattern (abbreviation of Monitoring, Analysis, Plan, Execution, and Knowledge). It has been widely used

[3-5] for designing self-managed systems that automatically adapt their structure and behavior based on the context changes. Adopting the autonomic computing for managing the patient treatment using the wearable technologies seems promising to detect the patient health anomaly and to provide personalized treatment that will be sent to the physician for validation [6].

Systems implementing real-world applications continuously evolve and generate unforeseen requirements [7]. Thus, conceiving only one MAPE-K loop is not sufficient to manage the evolution of context changeability. For instance, in healthcare, we consider managing patients with diabetes as a primary requirement. During treating diabetes, because of aging and biological changes, a new requirement such as managing the hypertension may occur. Thus, new processes for monitoring and controlling hypertension need to be integrated into the system. However, traditional management systems are considered as ad-hoc systems –designed and implemented from scratch, which impedes the dynamic self-management and requires additional human efforts to integrate and activate these processes. To enable the smart manageability, the system should be able to support the dynamic discovery of the management processes, their composition and dynamic coordination to manage more complex situations, especially those unpredictable at design-time. We refer to management processes any process that can control the context changeability of the system. The purpose of these processes is to monitor the system status, detect anomalies and/or generate recommendations. A management process can be the combination of existing management processes. The autonomic computing functions (Monitoring, analysis, Plan and Execution) are considered in our work as management processes. A smart IoT-based system requires also the integration of cognitive capabilities that allows not only understanding and perceiving the meaning of the received data but also the features of management processes in order to dynamically generate the adequate combination of processes to manage complex requirements.

Consequently, to facilitate the selection of the appropriate management processes, we defined in this paper four maturity levels for the development of smart IoT-based systems. These maturity levels definea set of the automated management processes that should be implemented based on the system requirements. For each maturity level, we defined a software design pattern that delineates the interaction among these management processes with the knowledge component and the human in order to enable the cognitive management with minimal human intervention. In this paper, we mainly detail the *Autonomic Cognitive Management pattern* which its implementation defined the most mature level. This pattern describes the coordination of the management processes and the ability of the system to dynamically discover new processes that can be combined at runtime with existing ones to manage unpredictable requirements. The ultimate objective is to provide flexible and smart IoT-based system that automatically adapts its processes based on the business context changeability.

This paper is organized as follows. Section 2 highlights the trend of wearable computing in healthcare. Section 3 discusses existing IoT platforms and their ability to manage the system context changes. Section 4identifies the maturity levels that we propose for the design and development of smart IoT-based systems. Then, Section 5delineates the *autonomic cognitive management pattern* for smart IoT-based system manageability. Section 6 illustrates the utility of the proposed pattern when managing a patient with diabetes and hypertension diseases (comorbidity management). Finally, conclusion and future work are presented in Section 7.

## 2 WEARABLE COMPUTING TREND IN HEALTHCARE

The wide adoption of wearable technologies propels industries and researchers to team up together and provide more efficient solutions to track the human activities and continuously monitoring the patient vital' signs [8-11]. Their integration in healthcare to monitor patients with serious conditions contributes to potentially reducing the healthcare cost by 88%[1]. Recently, Penders

[1] http://healthcare.orange.com/eng/news/latests-news/2014/infographic-wearable-tech-boom-in-healthcare

et al. [12] have pointed out the importance of tracking the lifestyle behaviors including physical activity, sleep, stress, diet, and weight management based on wearable sensors during pregnancy. The objective is to adapt and personalize the life style behaviors based on the collected data to provide healthier pregnancy. The use of wearable devices also pinpoints its impetus in the Active and Assisted Living (AAL) area, which aims at helping the disabled persons and elderly to offer a better quality of life. For instance, the work of Nicolelis [13] proposed to connect the brain to external devices in order to transform the brain signals into actions executed by the machine such as moving the limbs just by thinking [1] to help people suffering from catastrophic body paralysis performing the desired action[2].

Nowadays, the wearable market focuses on producing a new range of tiny wearables embedded within clothing and accessories to provide more efficient services and offer an easy interaction with. For instance, Ford is collaborating with RWTH Aachen University to integrate heart-monitoring sensors in the car seats to detect abnormal heartbeat and heart attacks. If detected, automated steering and braking systems will be activated[3]. Furthermore, Google X research lab has collaborated with the pharmaceutical business Novartis and Alcon's to create smart contact lenses[4] that measure glucose levels in tears for diabetes patients and correct vision for people with presbyopia. Other research activities focus on managing IT challenges related to the integration of IoT. For instance, IBM Watson Health and Apple have announced a new collaboration that focuses on providing cloud-based platform for a secure management of the patient data. Based on Apple ResearchKit, IBM's secure cloud and advanced analytics capabilities provide additional tools to accelerate discoveries across a wide variety of health issues[5]. At the University of Southern in Los Angeles, computer scientists and medical experts collaborate together and created an algorithm that uses data generated by various sensors including body sensors to better treat Parkison's disease [14]. In this way, medical experts can evaluate the treatment efficiency and notify patients.

The integration of wearables requires strengthening healthcare information systems in order to provide more personalized and high quality of care to patients. These technologies spawned a growing of the dataset and emphasized not only the huge volume of data but also its diversity and the speed at which it must be managed. According to the International Data Corporation (IDC)[6], the worldwide wearable market forecast is expected to reach 126.1 million units in 2019. The heterogeneity and the large amount of the generated data are challenging. It requires automating the data processing in order to facilitate the interaction with experts, while guaranteeing the scalability of the system and optimizing the processing cost in terms of allocated resources and response time. Moreover, in complex systems, the system requirements and context evolution may be unpredictable at design time. Thus, the coordination and the integration of these management processes such as new sensors to monitor the system status are required. Thus, it is important to provide a flexible architecture able to integrate new processes that cooperate with existing processes to generate new knowledge concerning the patient and/or to manage detected/predicted anomalies.

In the next section, we discuss existing works dealing with the integration and the management of IoT-based systems.

## 3 RELATED WORK

With the recent integration of IoT, different research activities focused on proposing platforms and architectures to manage IoT-based systems in various domains. IoT-A [15] is an IoT reference architecture that has been proposed to enable the interoperability among IoT connected devices. It describes sensors properties based on Semantic Sensor Network Ontology (SSN) [16], which is the

---

[2] http://www.ctvnews.ca/sci-tech/cyborg-soccer-how-a-paraplegic-took-first-kick-at-the-world-cup-1.1868837

[3] http://www.medtees.com/content/ecg_seat_fact_sheet_2.pdf

[4] http://www.cnet.com/news/google-extends-smart-lens-tech-for-those-with-diabetes-vision-problems/

[5] IBM Press: http://www-03.ibm.com/press/us/en/pressrelease/46583.wss

[6] International Data Cooperation: http://www.idc.com/getdoc.jsp?containerId=prUS25519615

most used ontology describing sensors and devices proposed by the W3C, the service model with OWL-S and extends them with an IoT information Model [17]. Moreover, IoT-A integrates the cloud computing for complex event processing [18] to guarantee scalability and efficiency.

OpenIoT [19] is a semantic cloud-based approach for implementing and integrating IoT solutions. It uses X-GSN to annotate the sensors and observed value based on SSN ontology, and stores them in RDF stores in a cloud infrastructure in order to guarantee the scalability and the elasticity of the platform. Based on the semantic annotation, OpenIoT enables the semantic search and discovery of sensors and services. The observed data is stored as linked data and processed based on SPARQL queries which are continuously executed once data arrive. Mobile devices are connected to the X-GSN middleware via publish/subscribe Mobile broker in order to guarantee near-real time management.

Mingozzi et al. [20] proposed the Building the Environment for the Things as a Service (BETaaS) for the integration of distributed and heterogeneous existing IoT systems. The solution adopted in BETaaS concentrates on exposing things as services (TaaS) through service-oriented interfaces, thus the integration is achieved with limited efforts and modifications. BETaaS is a semantic-driven solution where two ontologies [21] are defined: the BETaaS Things Ontology which reuses existing ontologies such as SSN, OWL-Time and QUDT, and the BETaaS Context Ontology which is the integration of the BOnSAI ontology, GeoNames ontology and GeoSPARQL ontology. BETaaS[22] includes also a big data manager in the TaaS and service layers that have the main functionalities gathering, storing, adapting, processing, and analyzing data. BETaaS is conceived to the management of smart building.

In healthcare domain, Lasierra et al. [23] proposed an ontology to describe the patient’s vital signs and to enable semantic interoperability when monitoring patient data by formalizing the X73 standard. Following the same direction, Kim et al. [24] proposed an ontology driven interactive healthcare with wearable sensors (OdIH_WS) to acquire context information at real time using ontological methods by integrating external data such as meteorological web site in order to prevent disease. Forkan et al. [25] proposed a cloud-based context-aware system called CoCaMAAL which covers challenges related to data collection and data processing in ambient assisted living systems. The authors proposed to mitigate the complexity of data computation from sensors to the cloud. They identified an abstract ontology to describe the context including patient information, the environment and devices. Jiang et al. [26] are interested in big data solutions for wearable systems in healthcare. They proposed a wearable sensor system with an intelligent information forwarder that adopts the Hidden Markov Model (HMM) to estimate the hidden wearer’s behaviors from sensor readings, and to determine the probability of the patient has a specific health state.

Based on the state of the art, we noted that these works mainly address the heterogeneity of the IoT-generated data, the big data management and scalability management. Almost works are using ontologies to describe sensors and their properties. However, an IoT-based system includes also actuators which are considered as the entry point to manage the system based on the monitored and analyzed data. Moreover, there is a lack of self-management properties that allows the system automatically detecting anomalies, planning and performing actions to manage the system business context evolution. In this context, Ben Alaya et al. [27] have proposed the FRAMESELF framework implementing the autonomic computing paradigm for the self-management of M2M systems in the context of smart cities. Through implementing a rule-based approach, FRAMESELF detects the context changes based on the collected data from sensors, plans for new actions and performs them through the actuators. These management processes are referring to a knowledge base describing the sensors and actuators. In order to manage the heterogeneity of devices, the authors extended their work with the definition of the IoT-O [28] ontology, which enriches and reused existing ontologies such as SSN, DUL, HREST, ACT, TIME, MSM and QUDT, to semantically describe sensors and actuators for the autonomic management of M2M systems. IoT-O has been instantiated for the management of smart homes.

Despite the diversity of the proposed platform, self-manageability properties of the IoT-based systems are rarely introduced, except the work of Ben Alaya et al. [27] which adopted the autonomic computing to automatically manage the system context changeability. However, the authors did not consider the coordination of the management processes in complex systems where multiple MAPE-K loops are required to manage the system requirements evolutions. Such situation is quite presented in healthcare, especially when managing the treatment of patients with chronic diseases based on the collected wearable data in order to predict and prevent health complications. It is important that the system should be aware about the context evolution of the patient health, and proposes to the experts at the right time the right processes to activate. To this end, we propose in this paper to deal with management processes coordination based on the system requirement evolution. Thus, we defined four generic maturity levels, independent from the applicative domain, for the development of smart IoT-based systems within organizations. These levels are detailed in the next section.

## 4 AUTONOMIC COGNITIVE IOT-BASED SYSTEM MATURITY LEVELS

Integrating new processes to meet new requirements requires additional efforts to understand the system process and avoid errors. It is important to provide a methodology that identifies and refines the system process improvement. In this context, the maturity levels have been introduced in order to define an evolutionary plateau for organization process improvement. Each maturity level comprises a set of process goals that, when satisfied, stabilize an important component of the software process [33].

IoT-based systems witness a rapid adoption in organizations. According to Gartner, more than half of major new business processes and systems will incorporate some element of IoT by 2020[7]. Thus, maturity levels for the development and improvement of IoT-based system processes are required. Some efforts have been invested in this area. Capgemini Consulting defines three maturity levels[8]: the basic information support which implements processes that deliver alerts and notifications on the product status; the remote operability support which implements processes that remotely configure the product; and the performance improvement support which implements processes that predict maintenance and enhance productivity based on sensor data. However, these maturity levels do not consider the context changeability and the ability of the system to discover new processes that meet the system requirements' evolution. Consequently, we propose in this paper four maturity levels implementing a set of processes that incrementally improve the system smartness. These processes are based on the combination of two principals: the autonomic computing and the cognitive computing in order to provide smart IoT-based systems.

The maturity levels that we propose enriched the autonomic computing maturity levels[9] (managed, adaptive, predictive and autonomic levels) with cognitive capabilities to represent the timeline evolution of the smart IoT-based systems development, as portrayed in Fig.1. The proposed maturity levels are the following:

The *Cognitive Monitoring Management* level is the basic level that allows IoT-based system interacting with human through collecting and visualizing the observations. At this level, only the monitoring process is automated and it implements cognitive capabilities that allow perceiving the received data streams. This level is adequate for near real-time visualization of the system context evolution. For example, in healthcare, it is important to continuously monitor the glucose level of prediabetes or elderly people, who have the risk to develop diabetes, and automatically detect possible degradation.

[7] http://www.gartner.com/newsroom/id/3185623
[8] https://www.capgemini-consulting.com/resource-file-access/resource/pdf/the-internet-of-things.pdf
[9] http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf

The *Predictive Cognitive Management* level is an evolution of the Cognitive Monitoring Management level. At this level, the system goes further the visualization and detection to apply intelligent mechanisms such as machine learning and data mining algorithms that allow the system learning and predicting other related anomalies initially imperceptible from existing observable parameters. For example, if the patient has confirmed diabetes and is following specific treatments to manage her glucose level, it is important to predict the hypertension as it is a risk factor of diabetes, especially if the patient is not equipped with a blood pressure sensor.

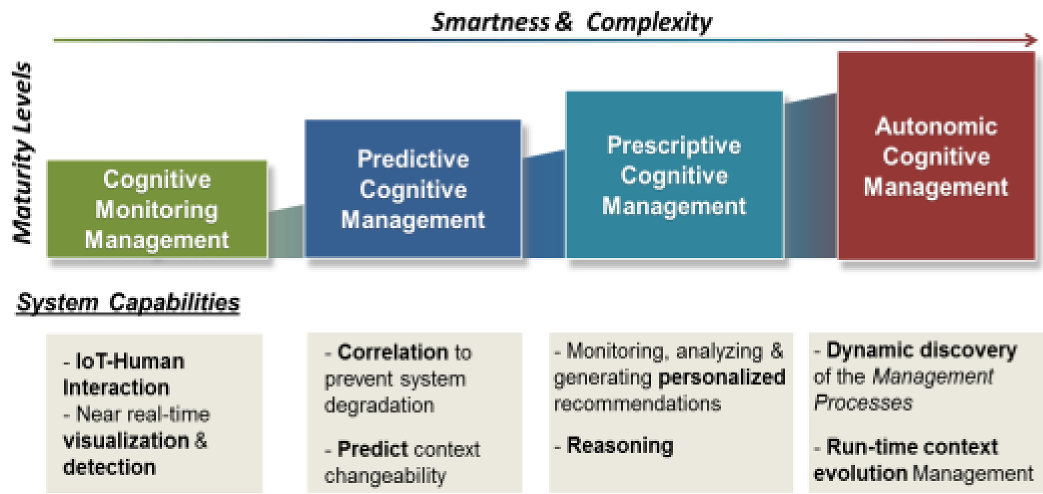


**Fig.1. Autonomic cognitive IoT-based system maturity levels**

At the *Prescriptive Cognitive Management* level, the system is able to provide business decisions to the expert based on the system context and through reasoning on the procedural knowledge, which includes the business rules populated by the business experts. As decisions are business related, the system sends the recommendations to the appropriate practitioner. For example, if the system detects that a patient with diabetes is getting worse while following a specific treatment; it will adapt the treatment through generating another personalized treatment and send it to the right physician.

The most mature level is the *Autonomic Cognitive Management* level. In complex systems, the context is dynamically changing, especially if the subsystems are interconnected and dependent. Thus, the system should be able to dynamically discover management processes based on the sub-systems' context evolution in order to provide a proactive management. For example, we consider a patient who is managed through ($M_d$, $A_d$, $P_d$, $E_d$) deployed for diabetes management. Because of aging, the patient may develop hypertension, thus, it is important that the system should be able to automatically search and activate the appropriate management processes managing the hypertension disease, besides diabetes, while interacting with experts who validate their activation as well as their recommendations for automatic execution.

We associated for each level a pattern defining the coordination of the management processes. In the next section, we mainly delineate the pattern associated to the most mature level which enables the smart manageability of the IoT-based systems.

## 5 AUTONOMIC COGNITIVE MANAGEMENT PATTERN FOR SMART IOT-BASED SYSTEMS

The *Autonomic Cognitive Management* pattern answers the following question: "*How a complex IoT-based system is able to dynamically manage the context changeability which may be unpredictable at design time?*" We followed a pattern template to describe the context, the problem that this pattern dealt with, the proposed solution and the consequences of its application.

***Context***. A smart IoT-based system is a system that should be able to manage its changes and evolution at runtime. The dynamic evolution of the business context requires dynamically integrating the management processes implementing business logic (such as new sensors for the monitoring, new analysis processes, etc), while interacting with the experts. Managing a complex system refers to simultaneously managing its sub-systems and coordinating the actions and their side effects that may impact the system functioning and state evolution, while optimizing the system design cost.

***Problem***. Management processes (Monitoring, Analysis, Plan and Execution) may be heterogeneous and distributed, which hinder the coordination and the collaboration of these processes to manage complex requirements at runtime. Moreover, in business-oriented applications such as healthcare, the interaction with experts is required to automate the execution of the generated treatment plans as well as to learn business rules for the adaptation. Furthermore, in complex systems, the massive deployment of management processes may lead to an increased cost. It is important to think about the ability of sharing and reusing processes such as the Analysis and the Plan processes in order to help reducing the cost.

***Solution***. To ensure the coordination of the management processes, we referred to the blackboard pattern [29]. This pattern has been widely used in the Artificial Intelligence domain for the dynamic control and coordination of the knowledge sources (KS) based on a control component (C). The control component supervises the *shared blackboard* among the KS. If the blackboard is modified, the control component activates the appropriate KS.

In our pattern, we consider that the management processes and the human experts are KS and we introduced a set of control components that coordinate the execution of these management processes based on the shared blackboard. We decomposed the Blackboard component into 4 sub-components: the *SensoryKnowledge* that semantically describes the used devices and the meaning of the generated data; the *DataCuratedBlackborad* that stores the monitored data; the *ContextKnowledge* that semantically describes the target goals of the monitored parameters and time-related information of each sub-system; and the *ProceduralKnowledge* describing the know-how for decision making. Each management process refers to these sub-components to read or write information. We identified also four control components controlling these different blackboards in order to coordinate the execution of the management processes. The proposed pattern coordinates the following interactions: Monitoring-Analysis, Analysis-Analysis, Analysis-Plan, Plan-Expert, Expert-Execution. For instance, the *ControlAA-AP* is observing the *ContextKnowledge*. If the context has changed, the controller activates the appropriate *Analysis* process. If an anomaly is detected, it activates the *Plan* process to generate recommendations.

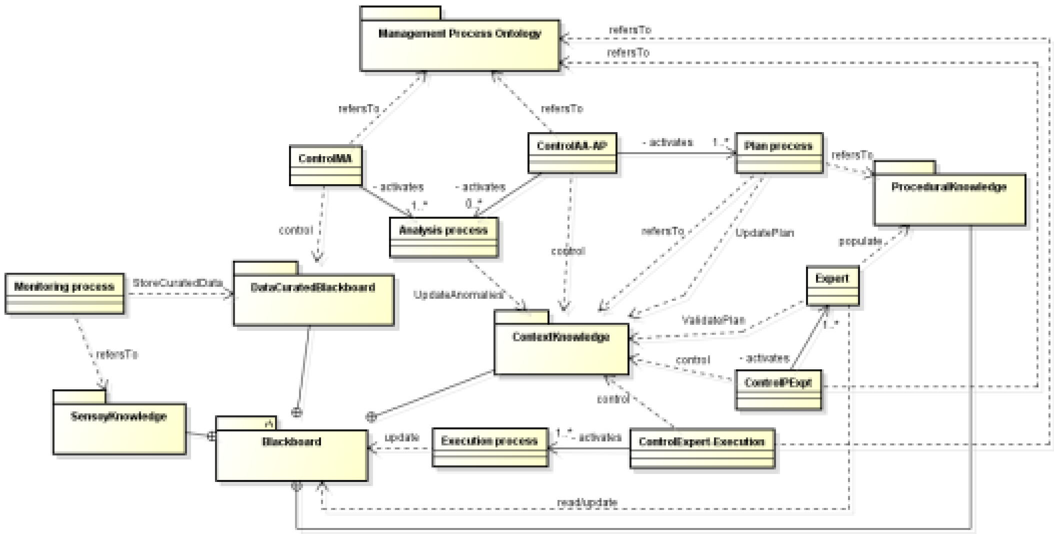


**Fig. 2.Autonomic cognitive management pattern**

To ensure the dynamic discovery and activation of the management processes in order to meet the system's requirements evolution, we propose to extend the basic control components with a semantic model, named *Management Process Ontology (MPO)*, describing the management processes as well as their conditions of activation, as presented in Fig.3. The "Management Process" class represents a generalization of the Monitoring, Analysis, Plan and Execution processes. The coordination is achieved through the control components that automatically discover the management process that should be activated to meet the system context evolution.

Fig.3 delineates the *Management Process Ontology*. We associated for each managed element a set of conditions describing the context, and a set of management processes managing its context changeability. Each management process is considered as an atomic process that has a set of preconditions expressed as conditions in order to guarantee its activation and enactment. These management processes can be the "mpo:MonitoringProcess", the "mpo:AnalysisProcess", the "mpo:PlanProcess" or the "mpo:ExecutionProcess". As we consider all processes are atomic, so the "mpo:MonitoringProcess" monitors only one "mpo:Parameter" and stores the measured data in the "mpo:DataBlackBoard" identified through an ID and an endpoint that will be used to retrieve the stored data. In case of observing more than one parameter, the "mpo:ManagedElement" is supervised by multiple "mpo:MonitoringProcess".

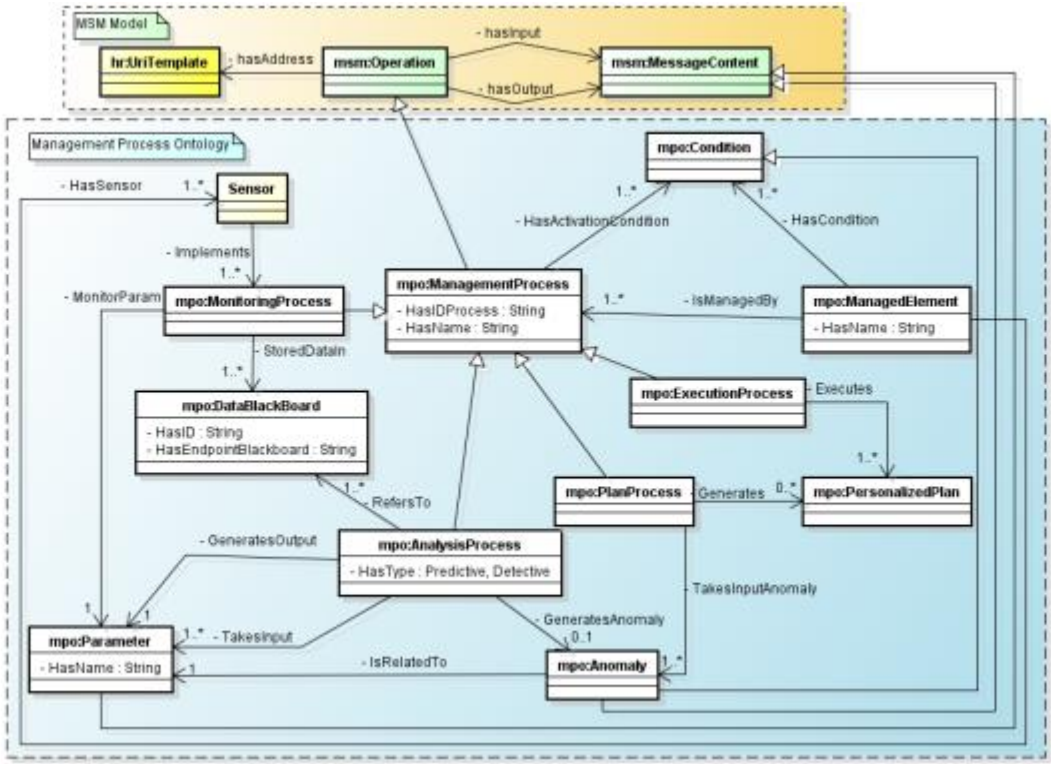


**Fig. 3.Management Process Ontology (MPO)**

MPO reuses existing ontologies such as the Minimal Service Model(MSM) ontology[10]which is a lightweight approach to the semantic modeling of Web service descriptions. Thus, each management process is a sub-class of the "msm:Operation" that has inputs and outputs represented respectively through the "msm:hasInput" and "msm:hasOutput" properties.MPO specializes the "msm:hasInput" and "msm:hasOutput" by introducing the following properties to guarantee the consistency of the acquired knowledge:

- "mpo:TakesInput" to specify that the analysis process takes as input at least one parameter.
- "mpo:GeneratesOutput" to specify that the analysis process analyzes one parameter. "mpo:TakesInputAnomaly" to specify that the plan process takes as input at least one anomaly.
- "mpo:Generates" to specify that the plan process generates a personalized plan.

The MPO has been conceived to allow the business experts populating the business rules through creating relations among these classes. Thus, the system easily interprets the business logic and enables the self-provisioning through dynamically discovering and composing the management processes to meet the system evolution. To this end, a set of SPARQL queries are proposed to enable the dynamic discovery of the management processes and keep the control components up-to-date with new management processes. Table 1 represents an example of a generic query that aims at discovering the monitoring processes and activating them based on the context changes. For instance, a patient with diabetes needs to check each 3 months his/her hypertension. Thus, the system should be able to search for possible available monitoring process to activate it. Thus, if the

---

[10]http://iserve.kmi.open.ac.uk/ns/msm/msm-2014-09-03.html

patient is equipped with a sensor that measures the blood pressure, the system will activate this process. Else, if the patient does not have a sensor, the control component executes the query presented in Table 2 to enable the predictive cognitive management of the hypertension. If the sensor and the monitoring process are activated, the system seeks for the appropriate analysis process in order to detect anomalies through enacting the Query3 presented in Table 3.

**Table 1. A query implemented in the ControlMA to activate the monitoring and sensor**

*Query1. Discovery of the monitoring process and sensor to be activated based on the context changes*

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: http://www.w3.org/2001/XMLSchema#
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX MPO:<http://homepages.laas.fr/emezghan/untitled-ontology-77#>
SELECT distinct (?el AS ?ManagedElement) (?s AS ?Sensor) (?ap AS?MonitoringProcess ) (?endp AS ?Endpoint )
Where
{
        ?apMPO:HasEndpoint ?endp.
        {
        ?ap1 rdf:typeMPO:MonitoringProcess.
        ?el MPO:IsManagedBy ?ap1.
        ?el MPO:HasSensor ?s.
        ?s MPO:Implements ?ap.
        Filter (?ap1 != ?ap).
        }
                {Select ?e ?ap Where
                {
                ?aprdf:typeMPO:MonitoringProcess.
                ?el MPO:HasCondition ?cond.
                ?apMPO:HasActivationCondition ?cond.
                }
                }
} GROUP BY ?ap ?el ?s ?endp
```

Query 2 presented in Table 2 allows discovering the required predictive analysis processes that should be deployed in order to provide preventive management based on the context of the managed element. This query is a sub-query, having three nested queries. At a first stage, it selects the list of management processes that have common preconditions with the managed element context. Then, from the list of the returned processes, it filters those all preconditions are satisfied and should be activated. The next step will focus on selecting from the generated list the processes that can be enacted based on the availability of the patient monitored data. For example, if the selected analysis process takes as input two parameters while the managed element has only one monitored parameter, the process cannot be activated due to knowledge/data incompleteness.

**Table2. A query within the ControlAA-AP to activate a predictive analysis processes**

*Query 2. Discovery of the predictive analysis process that should be activated*

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX MPO:<http://homepages.laas.fr/emezghan/untitled-ontology-77#>
Select (?el AS ?ManagedElement) (?ap AS ?AnalysisProcess ) (?endp AS ?Endpoint ) WHERE
{ Filter (?m1 = ?m2).
```

```
?apMPO:HasEndpoint ?endp.
        {Select distinct ?el ?ap ?m1 (count(?p) AS ?m2) where
        {?apMPO:TakesInputs ?p.
                {SELECT distinct ?el ?ap (count(?param) AS ?m1) Where
                        {
                        ?apMPO:HasEndpoint ?endp.
                        ?ap1 rdf:typeMPO:MonitoringProcess.
                        ?el MPO:IsManagedBy ?ap1.
                        ?ap1 MPO:MonitorParam ?param.
                        ?apMPO:TakesInputs ?param.
                        ?apMPO:HasType ?type.
                        ?apMPO:generatesOutput ?paramO.
                        ?el MPO:IsManagedBy ?ape.
                        ?ape MPO:generatesOutput ?param1.
                        ?apMPO:HasType ?type.
                        Filter (?ap != ?ape).
                        Filter (?param1 != ?paramO).
                        Filter (?type = 'Predictive').
                        } GROUP BY ?el ?ap ?m1
                }
        } Group by ?el ?ap ?m1 ?m2
        }
{Select distinct ?ap WHERE
{Filter (?c1 = ?c2).
        {SELECT distinct ?ap (count(?cond) AS ?c1) (count(?cond1) AS ?c2) Where
        {
                {
                ?aprdf:typeMPO:AnalysisProcess.
                ?apMPO:HasActivationCondition ?cond.
                ?el MPO:HasCondition ?cond.
                }
                UNION
                {
                ?apMPO:HasActivationCondition ?cond1.
                }
        } GROUP BY ?ap ?c1 ?c2}
}}}
```

***Consequence***. Reusing management processes such as the Analysis and the Plan process reduces the cost of deploying for each managed element its own management processes. Based on the semantic description of the management processes and the flexible implementation of the business rules, the proposed pattern enables the dynamic discovery and activation of the management processes. The proposed pattern also keeps the experts in the loop for decisions approval.

Conflicts when generating plans may occur when two or more Plan processes managing dependent sub-systems are simultaneously operating. As example of conflicts in comorbidity management is the presence of at least two treatments, identified by two plan processes, where the side effect of one of them represents a contraindication of the other; or also if one of them includes a drug that may interact with another drug belonging to the other treatment. We noted that when adopting a preventive approach, the simultaneous planning is seldom encountered. Thus, we proposed an ontology-driven approach [6] that enables the plan processes generating the appropriate treatment based on the diseases' risk factors, the treatment side effects, and patient medical conditions. Based on inference rules and reasoning techniques, ontologies ensure the consistency of the generated treatments.

In another context where ontologies are not used, standard solutions such as the synchronization based on tokens may be explored.

**Table3. Query within ControlMA for the detection analysis process activation**

*Query 3. Discovery of the detection analysis process that should be activated based on the new deployed monitoring process*

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX MPO:<http://homepages.laas.fr/emezghan/untitled-ontology-77#>
Select (?el AS ?ManagedElement) (?a AS ?AnalysisProcess) (?endpAS ?Endpoint) WHERE
{
Filter (?c1= ?c2).
?a MPO:HasEndpoint ?endp.
{
Select distinct ?el ?a (count(?param2) AS ?c1) (count(?param3) AS?c2) WHERE
{
        {
                {
                        {?m2rdf:typeMPO:MonitoringProcess.
                        ?el MPO:IsManagedBy ?m2.
                        ?m2 MPO:MonitorParam ?param2.
                        ?a MPO:TakesInputs ?param2.
                        }
                Union
                        {?aMPO:TakesInputs ?param3. }
                }
        }
        {SELECT distinct ?el ?a WHERE
        {
        ?el MPO:HasSensor ?s.
        ?m rdf:typeMPO:MonitoringProcess.
        ?s MPO:Implements ?m.
        ?el MPO:IsManagedBy ?m.
        ?m MPO:MonitorParam ?param.
        ?a rdf:typeMPO:AnalysisProcess./
        ?a MPO:HasType ?type.
        ?a MPO:generatesOutput ?param.
        ?a1 rdf:typeMPO:AnalysisProcess.
        ?el MPO:IsManagedBy ?a1.
        Filter (?a != ?a1).
        Filter (?type = 'Detection').
        }
}
} Group By ?el ?a ?c1 ?c2}}
```

The main objective of this pattern is to delineate the interaction and coordination of the management processes for the development of smart IoT-based systems. The instantiation of this pattern should be deployed in a semantic cloud-based big data platform in order to handle challenges related to the heterogeneity, big data and scalability management. In the new section, we propose to apply the proposed pattern for managing comorbidity.

## 6 USE CASE: COMORBIDITY MANAGEMENT

Comorbidity refers to the simultaneous or sequential occurrence of two disorders or illnesses in the same person. It also implies interactions between the illnesses that affect the course and prognosis of both [30]. In such cases, multiple management processes should be deployed and coordinated in order to manage the presented diseases and avoid possible complications derived from the diseases' interactions.

We assume that the proposed system is managing a group of patients with diabetes and another group of patient with hypertension. Each disease is managed by a set of management processes: for instance a patient with confirmed diabetes is managed by the *MonitorGlucose*, *AnalyzeGlucose*, *PlanDiabetes* and *Execution* processes; while a patient with confirmed hypertension is managed by *BloodPressureMonitor*, *AnalyzeHypertension*, *PlanHypertension* and *Execution* processes. Besides these processes, others such as the *PredictHypertension* are also defined and deployed. All these processes are annotated using MPO ontology.

Considering a patient, named Patient1, who is diagnosed with diabetes. He has a wearable that can measure blood sugar and blood pressure through two different interfaces implementing two measurement services. Initially, only the monitoring of the blood sugar is activated to continuously monitor the glucose level, and as there is no need to measure the blood pressure. Thus, the wearable may save the battery longer. In general, each 3 months, the blood pressure needs to be checked. Patient 1 is managed by the diabetes management processes. We propose to apply the proposed pattern in order to smartly manage the patient treatment and detect at early stage comorbidity.

Fig. 4 describes the behavior of our proposed system when implementing the cognitive management pattern to manage the Patient1 treatment. For clarity reasons, we abstract the control components, which are responsible of enabling the interaction among the management processes, and the blackboard sub-components (knowledge + databases). But, it is worth noting that in this use case the *SensorKnowledge* and the *ContextKnowledge* are represented using the *Wearable Healthcare Ontology (WH_O)* [31], while the *ProceduralKnowledge* is represented through the *Treatment Plan Ontology (TPO)* [6] which semantically annotates the medical interventions in order to enable the reasoning and generating personalized recommendations. The *WH_O* reuses: (i) the Sensor Model of IoT-O to describe the wearable capabilities, (ii) the Service Model to describe the wearable services and methods, and (iii) the Actuator Model for the autonomic management. An implementation of the cognitive monitoring management has been elaborated. For confidentiality reasons, the platform is not public. However, an evaluation of its performance as well as the list of technologies that have been used can be found in [32]. In this paper, we are interested in delineating the behavior of the system when implementing the autonomic cognitive management pattern, which is an extension of the cognitive monitoring management pattern, in order to discover and activate new processes based on the context changeability.

When receiving new glucose data, the *ControlAM* activates the *AnalyzeGlucose* process. If an increase of the blood sugar of Patient1 is detected after 2 months, the control component activates the *PredictHypertension* in order to check if the Patient1 presents an increase of the blood pressure to be taken into consideration when planning for new diabetes treatment.

If the hypertension is predicted, the system searches for the list of management processes to control hypertension and sends it to the physician for validation. In this case, the management processes are *MonitorBloodPressure* (because the patient is already equipped with a sensor measuring the blood pressure) and *AnalyzeHypertension*. Query1, presented in Table 1, represents a SPARQL query that deduces the activation of the *MonitorBloodPressure* based on the patient context, while query 3, represented in Table3, represents the SPARQL query that searches for reusable analysis process once the monitoring process is activated.

In parallel to these steps for deploying the hypertension management processes, the *PlanDiabetes* process takes into consideration the predicted hypertension, generates the appropriate treatment and sends it to the physician who validates or adjusts the recommendations.

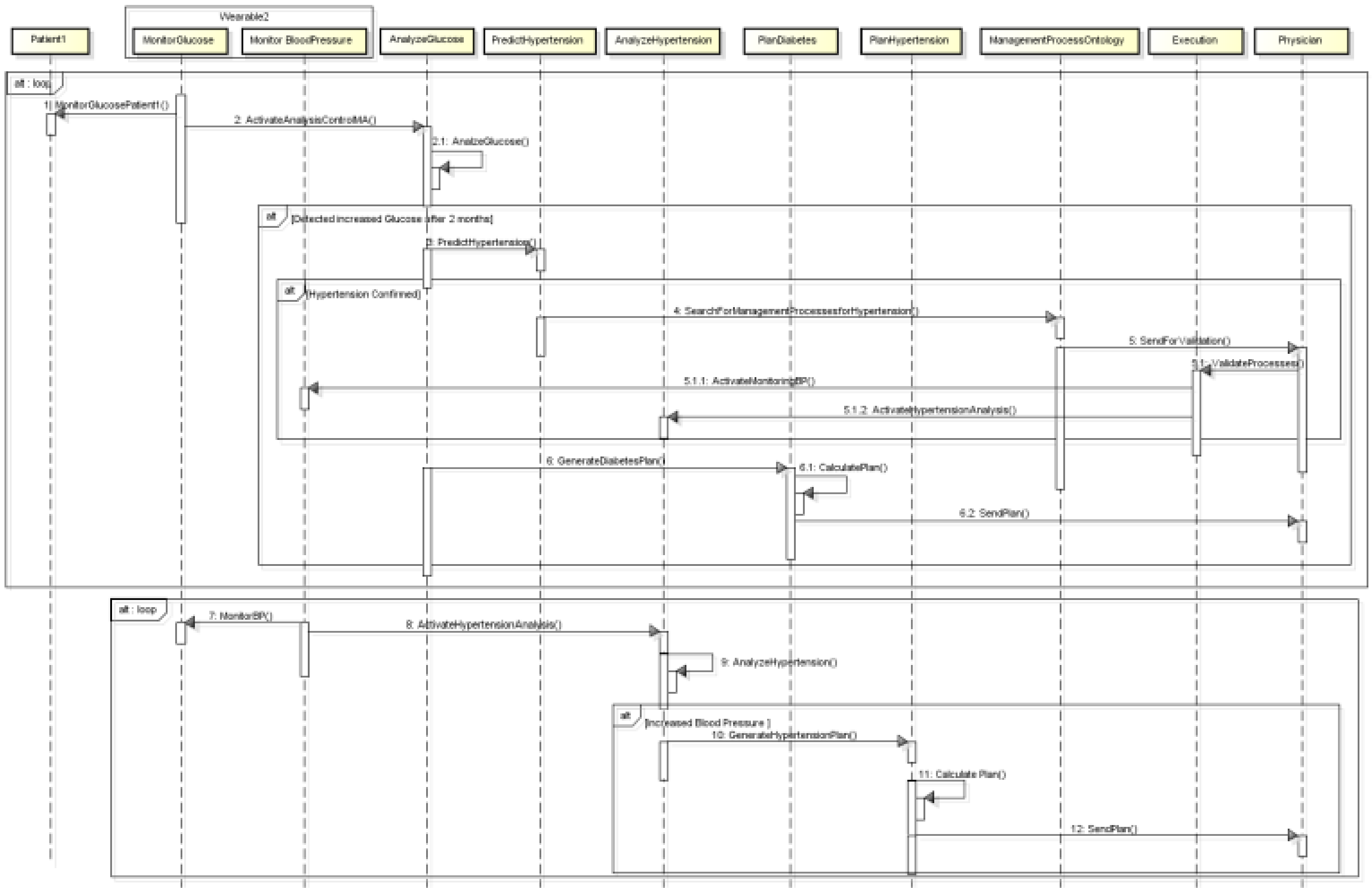


**Fig. 4. Instantiation of the autonomic cognitive management pattern in healthcare**

Finally, the Patient 1 will be managed by both the diabetes and hypertension management processes. The activation of these processes was driven by the Patient1 context changes; and dynamically composed based on the availability of data and the management processes.

## 7 CONCLUSION

By enabling communication and data exchange amongst heterogeneous devices, the IoT ultimately offer new opportunities for business development and/or accurate decision making. Many research activities have proposed IoT platforms to deal with such challenges. However, they rarely propose self-management properties that automate the system manageability and enable the continuous control of context changeability, as well as the coordination of the business processes to manage complex requirements.

To this end, we propose in this paper four maturity levels that define the different stages that an IoT-based system implements to reach the smart manageability. For each level, we define a design patter that integrates a set of autonomic and cognitive capabilities which are selected based on the system requirements. We delineated in this paper the *Autonomic Cognitive Management* pattern which defines the most mature level. The proposed pattern is generic (domain independent). It is not limited to manage the context changeability, but it coordinates also the business processes based on the collected data from IoT. We demonstrated the use of the proposed pattern in healthcare, in particular for comorbidity management.

Our future work focuses on deploying the instantiated management processes, the control and blackboard components within our Knowledge as a Service (KaaS) platform in order to evaluate the performance of the proposed solution in terms of response time and scalability management when discovery processes.

## ACKNOWLEDGMENTS

**This research is entirely funded by the National Research Fund (FNR) of Luxembourg under the AFR project.**

## REFERENCES

[1]: Nicolelis, M. A. “Mind in motion,” Scientific American, vol. 307, no. 3, pp. 58–63, 2012.

[2]: O. Kephart and D. M. Chess (2003). “The vision of autonomic computing. Computer”, 36(1), 41-50.DOI: 10.1109/MC.2003.1160055

[3]: M. C. Huebscher and J. A. McCann, “A survey of autonomic computing– degrees, models, and applications,” ACM Computing Surveys (CSUR), vol. 40, no. 3, p. 7, 2008.DOI:10.1145/1380584.1380585

[4] C. Klein, R. Schmid, C. Leuxner, W. Sitou, and B. Spanfelner, “A survey of context adaptation in autonomic computing,” in Fourth International Conference on Autonomic and Autonomous Systems (ICAS’08), pp. 106–111, IEEE, 2008.DOI: 10.1109/ICAS.2008.23

[5]: A. G. Ganek and T. A. Corbi, “The dawning of the autonomic computingera,” IBM systems Journal, vol. 42, no. 1, pp. 5–18, 2003.DOI:10.1147/sj.421.0005

[6]: Mezghani, E., Da Silveira, M., Pruski, C., Exposito, E., & Drira, K. An Ontology-driven Adaptive System for the Patient Treatment Management. in Twenty-Eighth International Conference on Software Engineering and Knowledge Engineering, pp. 329–332, Knowledge Systems Institute, 2016. DOI: 10.18293/SEKE2016-155

[7]: R. De Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel, et al., “Software engineering for self-adaptive systems: A second

research roadmap," in Software Engineering for Self-Adaptive Systems II, pp. 1–32, Springer, 2013.DOI:10.1007/978-3-642-35813-5_1
[8] E. Chiauzzi, C. Rodarte, and P. Das Mahapatra, "Patient-centered activity monitoring in the self-management of chronic health conditions," BMC medicine, vol. 13, no. 1, p. 1, 2015.DOI: 10.1186/s12916-015-0319-2
[9] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 1, pp. 1–12, 2010. DOI: 10.1109/TSMCC.2009.2032660
[10] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A review of wearable sensors and systems with application in rehabilitation," Journal of neuroengineering and rehabilitation, vol. 9, no. 1, p. 1, 2012.DOI: 10.1186/1743-0003-9-21
[11] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," IEEE Sensors Journal, vol. 15, no. 3, pp. 1321–1330, 2015.DOI: 10.1109/JSEN.2014.2370945
[12] J. Penders, M. Altini, C. Van Hoof, and E. Dy, "Wearable sensors for healthier pregnancies," Proceedings of the IEEE, vol. 103, no. 2, pp. 179–191, 2015.DOI**:** 10.1109/JPROC.2014.2387017
[13]: M. A. Nicolelis, "Brain–machine interfaces to restore motor function and probe neural circuits," Nature Reviews Neuroscience, vol. 4, no. 5, pp. 417–422, 2003.DOI: 10.1038/nrn1105
[14]: K. Pretz, "Better Health Care Through Data: How health analytics could contain costs and improve care." http://theinstitute.ieee.org/ns/quarterly_issues/tisep14.pdf, 2014. [Online; accessed 19-March-2017].
[15] : A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. Van Kranenburg, S. Lange, and S. Meissner, "Enabling things to talk," Designing IoT Solutions With the IoT Architectural Reference Model, pp. 163–211, 2013. DOI: 10.1007/978-3-642-40403-0
[16]: M. Compton, P. Barnaghi, L. Bermudez, R. GarcíA-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, et al., "The SSN ontology of the W3C semantic sensor network incubator group," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 17, pp. 25–32, 2012. DOI: 10.1016/j.websem.2012.05.003
[17]: N. Bui, "Project Deliverable D1.1 - SOTA report on existing integration frameworks/architectures for WSN, RFID and other emerging IoT related Technologies," 2014. Available at: http://www.meet-iot.eu/deliverables-IOTA/D1_1.pdf
[18]:S.Becher, T. Jacobs, C. Kleegrewe, S. Meissner, S. Meyer, G. Völksen, "Internet of things architecture iot-a project deliverable d2. 6," Available at http://www.meet-iot.eu/deliverables-IOTA/D2_6.pdf
[19]: J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte,M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko, et al., "Openiot:Open source internet-of-things in the cloud," in Interoperability andOpen-Source Solutions for the Internet of Things, pp. 13–25, Springer, 2015.DOI: 10.1007/978-3-319-16546-2_3
[20]: E. Mingozzi, G. Tanganelli, C. Vallati, and V. Di Gregorio, "An open framework for accessing things as a service," in Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on, pp. 1–5, IEEE, 2013.
[21]: I. Mendia, "Semantics in betaas," 2014. Available at:
http://www.betaas.eu/docs/Semantics%20in%20BETaaS-SanktAugustin.pdf
[22]: S. Kyriazakos, B. Anggorojati, N. Prasad, C. Vallati, E. Mingozzi, G. Tanganelli, N. Buonaccorsi, N. Valdambrini, N. Zonidis, G. Labropoulous, et al., "Betaas platform–a things as a

service environment for future m2m marketplaces," in Internet of Things. User-Centric IoT, pp. 305–313, Springer, 2015.DOI: 10.1007/978-3-319-19656-5_43

[23]: N. Lasierra, A. Alesanco, S. Guillén, and J. Garcia, "A three stage ontologydriven solution to provide personalized care to chronic patients at home," Journal of biomedical informatics, vol. 46, no. 3, pp. 516–529, 2013.DOI: 10.1016/j.jbi.2013.03.006

[24]: J. Kim, J. Kim, D. Lee, and K.-Y. Chung, "Ontology driven interactive healthcare with wearable sensors," Multimedia Tools and Applications, vol. 71, no. 2, pp. 827–841, 2014. DOI:10.1007/s11042-012-1195-9

[25]: A. Forkan, I. Khalil, and Z. Tari, "Cocamaal: A cloud-oriented contextawaremiddleware in ambient assisted living," Future Generation ComputerSystems, vol. 35, pp. 114–127, 2014; DOI: 10.1016/j.future.2013.07.009

[26] P. Jiang, J. Winkley, C. Zhao, R. Munnoch, G. Min, and L. T. Yang, "Anintelligent information forwarder for healthcare big data systems with distributedwearable sensors," IEEE Systems Journal, vol. 10, pp. 1147–1159,Sept 2016.DOI: 10.1109/JSYST.2014.2308324

[27]: M. BenAlaya, and T.Monteil. "FRAMESELF: an ontology-based framework for the self-management of machine-to-machine systems." Concurrency and Computation: Practice and Experience 27.6 (2015): 1412-1426.DOI: 10.1002/cpe.3168

[28]: M. Ben Alaya, S. Medjiah, T. Monteil, and K. Drira, "Toward semanticinteroperability in onem2m architecture," IEEE Communications Magazine,vol. 53, no. 12, pp. 35–41, 2015. DOI: 10.1109/MCOM.2015.7355582

[29]: N. I. on Drug Abuse and U. S. of America, "Comorbidity: Addiction and other mental illnesses,"

[30]: P. Lalanda, "Two complementary patterns to build multi-expert systems," in Pattern Languages of Programs, 1997.

[31]: E.Mezghani, E.Exposito, K.Drira, M.Da Silveira, & CPruski. (2015). A semantic big data platform for integrating heterogeneous wearable data in healthcare. Journal of medical systems, 39(12), 185.DOI: 10.1007/s10916-015-0344-x

[32]: E. Mezghani, E. Exposito and K. Drira. (2017). A Model-Driven Methodology for the Design of Autonomic and Cognitive IoT-Based Systems: Application to Healthcare. IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 1, no. 3, pp. 224-234 doi: 10.1109/TETCI.2017.2699218

[33]: M.Paulk (1993). Capability maturity model for software. Encyclopedia of Software Engineering.