2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing

Virtualization and Cognitive Management of Real World Objects in the Internet of Things

[1]Dimitris Kelaidonis, [2]Andrey Somov, [1]Vassilis Foteinos, [1]George Poulios, [1]Vera Stavroulaki, [1]Panagiotis Vlacheas, [1]Panagiotis Demestichas, [3]Alexander Baranov, [2]Abdur Rahim Biswas, and [2]Raffaele Giaffreda

[1]University of Piraeus,
Department of Digital Systems
Piraeus, Greece
e-mail: {dkelaid, vfotein, gpoulios, veras, panvlah, pdemest}@unipi.gr

[2]CREATE-NET
eNTIRE
Trento, Italy
e-mail: {asomov, arahim, rgiaffreda}@create-net.org

[3]Moscow State Aviation
Technological University
Moscow, Russia
e-mail: radio@mati.ru

Abstract—**This paper presents a framework for the virtualization of real world objects and the cognitive management of their virtual counterparts. The framework consists of three levels of functionality and each level comprises cognitive entities that provide the means for self-management and learning, allowing for smart, flexible applications and objects. The presented framework enables the abstraction of the heterogeneity that derives from the vast amount of diverse objects/devices, while enhancing reliability and facilitates the consideration of the views of various users/stakeholders (owners of objects & communication means) for ensuring proper application provision, business integrity and, therefore, maximization of exploitation opportunities. The paper also presents a corresponding prototype that has been developed for the validation of the proposed approach, in a real-life fire detection scenario in a Smart Home.**

Keywords- Internet of Things, Cognitive Management, Self-Management, Virtualization, Virtual Objects, Wireless Sensor Networks, Web of Things

I. Introduction

The term ‘Internet of Things’ (IoT) exists for more than 10 years [1]. However, a reader can find various definitions in the literature [2]. The common understanding of the IoT is associated with the future where all physical and digital objects or things could be interconnected by appropriate Information and Communication Technologies (ICT) to enable new applications and services [3].

The first attempts towards the virtualization of Real World Objects (RWO) were connected with Radio Frequency Identifiers (RFID) [4] which could capture only ‘raw’ data. Next steps were done in the contextualization of captured data [5]. Nowadays, Virtual Objects (VOs), i.e. virtual representations of objects/things, tend to be ‘smarter’ by enriching their models with cognitive management functions and user information [6]. Although much research has been done in terms of virtualization in the IoT domain, much more research is required to understand how things and their corresponding VOs can be connected and interact in a smart way, or how the concepts can be exploited and implemented.

In this direction, this work proposes a cognitive management framework for the IoT and presents a corresponding prototype implementation it. The virtualization of RWOs in the framework is realized through the creation of VOs which are semantically enriched with context related information. Additionally, VOs of different types can be combined in a more sophisticated way by forming Composite VOs (CVO), which provide services to high-level applications and end-users. The main features of the framework are the cognitive mechanisms which provide the means for self-management (configuration, healing, optimization, protection) and learning. In this respect, they allow for perceiving and reasoning on context (e.g., based on event filtering, pattern recognition, machine learning), conducting associated knowledge-based decision-making (through associated optimization algorithms and machine learning), and autonomously adapting the behavior and configuration of applications and objects according to the derived situation. A prototype implementation is presented through which the validation of the framework is done in a real scenario where two sensor networks supporting different technologies for fire detection are deployed in an apartment. The main idea is that the cognitive management framework provides recommendations to the sensor networks on the most appropriate nodes for fire monitoring in a given situation. The criteria for the recommendations are the energy saving and reliable fire monitoring.

The rest of this paper is structured as follows. Background on the IoT and Web of Things key concepts, virtualization of RWO in the IoT, and fire detection using WSN technology is presented in Section II. The main theoretical concepts as well as the description of the functional levels and the cognitive components of the proposed framework can be found in Section III. An indicative scenario for the application of this framework is discussed in Section IV. The implementation details are provided in Section V. Finally, the paper concludes with a summary of the main outcomes as well as targets for future extensions of this work.

978-0-7695-4865-4/12 $26.00 © 2012 IEEE
DOI 10.1109/GreenCom.2012.37

II. Background

A. Internet of Things (IoT) and Web of Things (WoT)

The vision of the IoT [7] is to enable things to be connected anytime, anyplace with anything and anyone ideally using any network and any service. The virtual representations of RWOs contain or link to information regarding their identification and the description. The RWOs can communicate information about themselves and can access information that is provided by other things, via the Internet. Additionally, there are various technologies that are considered in the realization of the IoT, such as RFID, Sensor Networks, Semantic Technologies, Machine-to-Machine (M2M) communication.

The *Web of Things (WoT)* concept proposes the integration of smart things and their services and counterparts on the web, which can be seen as an enhanced IoT concept [8]. As such, the WoT concept, increases the need for intelligence that will enable connectivity, internetworking, interoperability and relevance between the physical world and the corresponding virtual world resources.

This paper describes an approach for the virtualization of RWOs, their inter-connection, enhancement and efficient exploitation through the cognitive mechanisms of the proposed framework, that address the requirement for additional intelligence in the IoT (or WoT).

B. Virtualization of Real World Objects in the Internet of Things

A number of frameworks for the virtualization of RWO and their management have been proposed recently. In [9] the authors propose the IoT architecture applied mainly in the context of people with disabilities. The architecture has three layers: perception, network, and application. At the perception layer the real objects are identified and information about them is captured by sensors, smart phones, etc. The main goal of the network layer is to deliver gathered data using the wireless/wired networks and/or the internet to the application layer which enables an operation support platform. This platform, in turn, can be accessed by different applications and services. In contrast to our contribution, this work mainly deals with data collection, delivery, and sharing with its further use in different applications.

Other approaches try to address technical issues in the RWO virtualization (without addressing the cognitive aspects of the VO) and propose to embed the TCP/IP stacks in the devices [10].

There are also works in the IoT domain which enrich the VOs with context information [5] or propose techniques for RWOs [11] and their description [12]. However, none of the approaches considers 'smart' capabilities of the virtualized RWOs. This paper tries to fill this gap by proposing additional architectural and functional blocks which empower the VOs with cognitive functions (see Section V-D).

C. Fire Detection with Wireless Sensor Networks

In this work we use Wireless Sensor Network (WSN) technology (which is considered as one of the pillars of the IoT [1]) for a prototype implementation and validation of the proposed framework. To facilitate the understanding of the reader for the case study, we briefly introduce fire monitoring approaches with WSNs [13].

The first sensor nodes to appear, were simple wireless devices with temperature, humidity, luminosity, barometric pressure, and accelerometer sensors on board. Low power consumption of these devices (as opposed to gas sensors) [14] allowed researchers in [15][16] to deploy WSNs for fire monitoring using only temperature, humidity, and barometric pressure sensors. In [16], specially developed software assists a user in the establishment of the critical thresholds of environmental parameters considered for fire identification and notification. A system presented in [17] automatically processes the sensing data, calculates the fire risk-level index, and compares it with a threshold level.

The second approach is based on pyrolisys detection using a WSN [18]. Pyrolysis is a gas mixture of hydrogen (H_2) and carbon oxide (CO) which appears due to smoldering or overheating of objects before inflammation. The system presented in [18] has a gas sensor on board and is much more power 'hungry' with respect to approaches based on temperature, humidity and barometric pressure sensors [14]. At the same time the *pyrolisys* approach is much more reliable: the sensor's response is in milliseconds range and the fire can be predicted long before the inflammation.

The listed systems have a rather simple automation logic based on the comparison of the sensed values with their corresponding thresholds. In the present approach, the fire detection system is made smarter by devising the cognitive functions for the sensor node-VOs and by exploiting the Internet as a communication infrastructure in order to create a fire detection application.

III. Framework Functional Levels and Cognitive Components

The vast amount of everyday objects that are comprised in the IoT generates technological heterogeneity that should be addressed in order to achieve the integration between different types of hardware and software infrastructures. The abstraction of technology-specific details will enable the development of applications and interoperability of software entities that handle or use the managed devices. Additionally, one equally important issue is the different views and goals that derive from users and stakeholders of IoT systems.

In order to address these issues, this work proposes a cognitive management framework which consists of three levels of functionality (Figure 1), and is reusable for various and diverse applications. The three defined levels as well as the components of this framework are described in more detail in the following sub-sections.

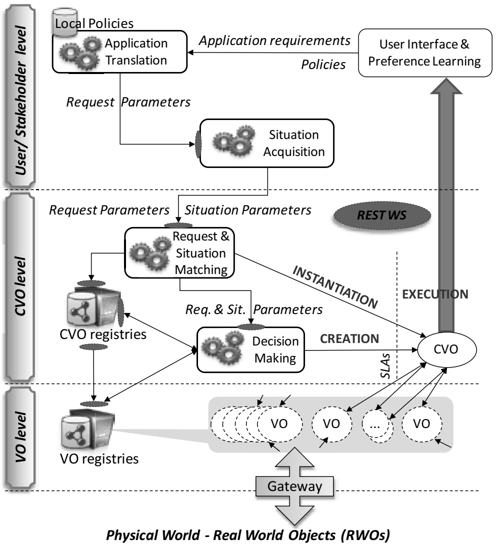


Figure 1: Cognitive management framework functional levels

A. Virtual Objects Level

The lowest level defined is the VO level and provides the bridge between the physical and the virtual world. It comprises the virtual representations (i.e. machine-readable descriptions) of the RWOs and the components that are used to store the VO-related information, namely the VO registries.

1) Virtual Objects: A VO is the dynamic virtual representation of a RWO that contains information for the description of the RWO, while it is implemented as computer software that is used to link the RWO with the virtual world. Any object, whether it embeds ICT capabilities (e.g. sensors, webcams, smartphones, etc) or not (e.g. persons, furniture, buildings, etc) is considered to be a RWO. An ICT object can be associated to a non-ICT object (e.g.: a temperature sensor that is associated to a room and measures the room temperature) and can be described in such a manner that it enables its exploitation and accessibility in the virtual world. Additionally, a non-ICT object can be implicitly described in the virtual world, through the association with the ICT object. Essentially, a VO provides an abstract representation of the features and capabilities, of both ICTs and non-ICTs in the virtual world.

2) VO Registry: Information regarding the available VOs is stored in distributed VO registries. An information model consisting of the appropriate attributes for the description of VOs has been designed and is depicted in Figure 2. It includes information regarding the ICT object being virtualized, its location in the real world expressed in geographic coordinates, its associations with any non-ICT object(s), information on how the VO can be accessed and used, as well as details regarding the functions that it offers. Regarding the latter, details on their inputs, outputs as well as various associated costs and utilities arising from their usage, are also stored in the VO registries. Such information on *function costs* and *function utilities* are taken into account when selecting the most appropriate VOs based on the applicaiton requirements and policies, i.e. during the decision making process, which will be described in more detail in the following paragraphs. Uniform Resource Identifiers (URIs) [19] have been used for the identification of VOs, ICT objects, and their offered functions.

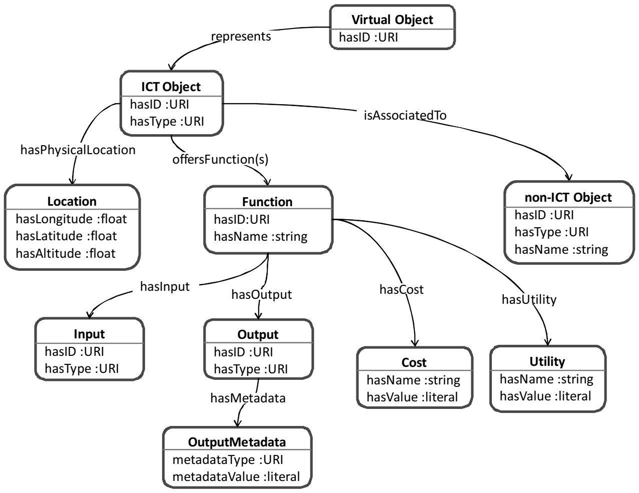


Figure 2: VO Information Model

B. Composite Virtual Objects Level

The CVO level bridges the users/stakeholders level with the available VOs in the VO level. Various building blocks are available in this level such as the CVOs and CVO registries as well as the cognitive entities for “Request and Situation Matching” and “Decision Making”.

1) Composite Virtual Objects: The CVOs are cognitive mash-ups of semantically interoperable VOs that are combined in order to offer services that will fulfil a requested application by a user/stakeholder. A CVO uses the functions that are provided by VOs and combines them for the provision of composite services. A CVO is self-managed and self-configurable, capable of being used also out of the context for which it was initially created. One or many CVOs may be used in order to compose an application providing a set of services in accordance with user/stakeholder requirements. Similar to the VOs, CVOs are described through semantically enriched information that is stored in the CVO registries.

2) CVO Registry: The information about CVOs, their components and the context under which they were created is stored in the CVO registry. More specifically, each CVO is created dynamically by the cognitive mechanisms that are comprised in the framework, based on a set of high-level user/stakeholder requirements taking into account also situational conditions. Therefore, the CVO registry also stores information on the requirements and context under which a CVO was created or deployed. Furthermore, the CVO-related information involves data regarding the user feedback on CVO. The cognitive mechanisms (e.g. Request and Situation Matching) can use the CVO registry records as knowledge on past CVO instantiations, enabling the direct deployment of a CVO.

3) Request and Situation Matching: Request and situation matching is performed to avoid re-creating CVOs that exist and can be re-instantiated under the particular context of a request. Vectors composed of numerical values and nominal categories represent each instance of a request as well as situation parameters. A similarity metric is defined, to be applied at any new such vector against the vectors stored in the CVO registry (i.e. records of request and situation parameters paired with the produced CVO). Similarity is then compared to a threshold, and in case this is exceeded, the CVO created for the corresponding request and situation sample is re-instantiated, resulting in performance gain and re-usability.

4) Decision Making: This entity is responsible for the creation of CVOs. These CVOs will deliver appropriate services according to the requested functionality, policies and the VOs that are available in the area of interest. Knowledge of these factors is necessary for the operation of the Decision Making. Additionally, it comprises both optimization and cognition aspects and it constitutes one of the main components of our framework's cognition loop. More details on the functionality of this component are provided in the following sub-sections.

a) Inputs: The Decision Making enity receives a set of available VOs, as well as a set of "Requested Parameters" (comprising the requested functions and provided policies). This information is provided by the Request and Situation Matching entity. Available VOs correspond to the VOs that exist ones in the specific area, at the certain time and that can be exploited for the composition of the CVO. The requested functions denote the functionality that this CVO should provide and they constrain the VOs that will be selected. Policies should be respected and also act as further constraints in the selection of the most suitable VOs. The Decision Making entity also retrieves further information on these available VOs from the VO registry. In particular, values for the "utility" and "cost" of each VO are acquired. Any attribute with a positive effect deriving from the operation of a VO can be considered as a utility value for the VO. On the contrary, attributes with negative effect are considered as cost values. This effect can be observed either at the VO itself either at the context of this VO. Indicative utility and cost attributes are "quality", "performance" and "security", for the first case and "energy", "network" and "expenditures" for the second one. Policies that the created CVO must respect are expressed as weights for the aforementioned attributes.

b) Outputs: The objective of the Decision Making Entity is to find the most suitable VOs to form an optimal CVO in terms of functionality and policies. Therefore, the output of its operation is the creation of a CVO, its registration in the CVO registry, its deployment and activation.

c) Optimization Process: Each available VO is characterised from a set of values. These values describe its features, which are categorized as utilities and costs. Utilities have a positive meaning, while costs have a negative one. These values are weighted according to the requested policies. In addition, the offered functions of the VO are correlated with the requested ones and the output of this procedure indicates the suitability of each of them. Hence, the aggregation of these values, in conjunction with the correlation of each function, provides an overall value of each function of each VO. The Decision Making evaluates these overall values through the exploitation of a mathematic formulation (objective function, constraints). The goal is to select VOs and functions of these VOs that have a maximum aggregated overall value. In this manner, the composed CVO is the most suitable according to the request.

C. Users / Stakeholders Level

This level focuses on the interface with users and stakeholders. It comprises functionalities and interfaces for acquiring user/stakeholder requests, preferences and constraints that are taken into account for the automated creation and dynamic configuration of CVOs, for the provision of smart, customized or personalized services. Users/stakeholders can provide a set of requirements and policies for the creation of an application that is needed. Furthermore, cognitive mechanisms are exploited in order to build knowledge on users based on provided preferences and feedbacks.

1) Application Translation: The role of this component is to perform a translation of high-level goals to lower level CVO and VO required functions. The high level user requirements and policies are mapped to service logic requirements and corresponding required CVO and VO functions that are forwarded to the Situation Acquisition mechanisms.

2) Situation Aquisition: The produced outputs from the Application Translation mechanisms pass to the Situation Acquisition mechanisms that infer relevant situational dimensions (e.g. location, time, VO states, etc). Through these mechanisms, the framework is able to acquire the description of the current situation and anticipate changes on it. The result of the situation acquisition process is forwarded to the Situation and Request Matching mechanisms.

IV. Scenario

A set of RWOs is installed in a smart home [20] where an elderly woman (Sarah) lives. Sarah needs continuous medical surveillance by the staff of a medical centre which monitors the environmental conditions inside her home. In this respect several sensors have been installed. In particular the first sensor platform includes temperature, humidity, and luminosity sensors, while the second one comprises the pyrolysis sensor to forecast the fire. It should be noted that temperature and humidity sensors can be *reused* for the fire detection application (see Section V.A.1). The sensors' reuse is supported by the framework's cognitive functions. In addition, a set of actuators (fan, coloured lamp, light emitting diode, and buzzer) have been installed in Sarah's smart home to react on the environmental events.

The scenario consists of five different phases. The first phase comprises the provision of high level of requirements and policies that are provided by a member of staff through an appropriate user interface to the Application translation mechanisms. Then the Situation acquisition mechanisms add a set of situation parameters on the application requirements and send these to the "Request and Situation Matching" mechanisms. These mechanisms, search in the CVO registry for a created CVO that could fulfil the requested application. If an appropriate CVO already exists it is deployed / instantiated directly. Otherwise if a matching CVO is not found, the Decision Making mechanism is triggered, in order to select the appropriate VOs to satisfy the requests and policies. Information on the created CVO along with the request and situation parameters as well as information on the VOs, comprised in the CVO, is stored in the CVO registry.

V. Framework Implementation

A. *Real World Objects: Hardware and Communication*

In this implementation we use three embedded platforms (in our case RWOs): WaspMote [21], a gas sensor node [14] for fire detection and Arduino [22] for the management of actuators. A laptop serves as a gateway and the Internet as the communication infrastructure among sensors, actuators and the cognitive management framework. The platforms and their interconnection are described in more details in the following sub-sections.

1) Sensing: For fire sensing two platforms are used: WaspMote and gas sensor node. WaspMote (see Figure 3a) is a commercial wireless sensor platform designed by Libelium. The platform advantages include its extensibility and ease of use. Extensibility is provided by a number of extension boards (with various sensors on board) for various applications: smart city, smart parking, agriculture just to list a few. One extension board can be connected to the main sensor board at a time. The main sensor board contains the necessary components for sensing: microcontroller Atmega1281 by Atmel, wireless communication unit XBee-802.15.4 (ZigBee, 2.4 GHz), temperature sensor and accelerometer, power supply provided by Li-ion batteries. For our scenario the 'Smart City' extension board with humidity sensor has been used. In total the temperature and humidity sensors for fire detection have been used. In fact, to detect the fire using these sensors the increase of temperature and decrease of humidity need to be checked at the same time. The sensors' power consumption is less than 1 mW in active mode.

The gas sensor node (see Figure 3b) is a custom design platform for the detection of hazardous gases. The platform supports semiconductor and catalytic sensors. In this work we use a semiconductor one. The principle of fire detection is based on the sensing of pyrolysis in the environment. Pyrolisys is a gas which contains hydrogen (H_2) and carbon oxide (CO). This gas appears due to smoldering or overheating of materials that help to predict possible fire in advance. The sensor node is built on Atmega168p microcontroller by Atmel, wireless modem ETRX2 (ZigBee, 2.4 GHz) by Telegesis. Power consumption of the node is, however, around 150 mW which results in approximately 1.5 years of autonomous operation. More details on the design and operation of the node can be found in [14][23].

Both sensor platforms exploit wireless communication for connection with the laptop.

2) Actuation: To alarm a user about the potential dangerous situation we use three actuators (associated with the WaspMote sensors) connected to Arduino (see Figure 3c): red Light Emitting Diode (LED), coloured lamp, and fan. The coloured lamp alarms a user when humidity

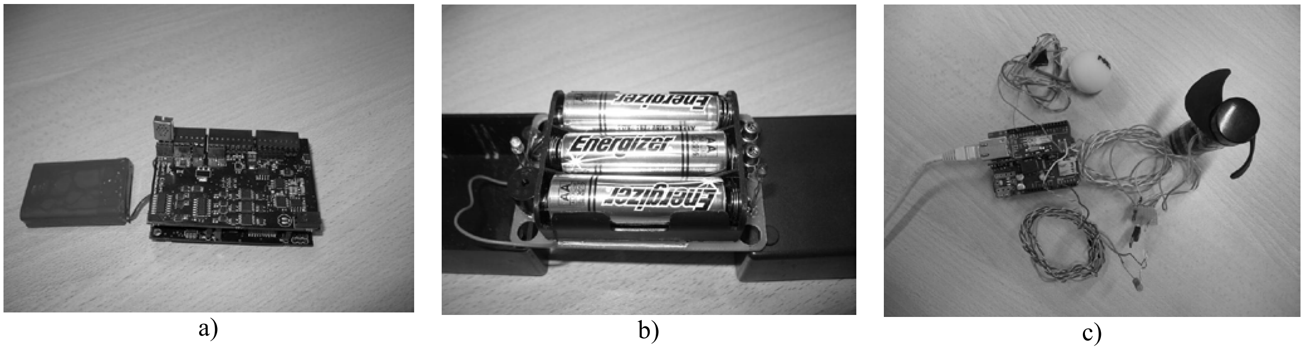

a)
b)
c)

Figure 3: Sensing and actuation devices used (a) WaspMote wireless sensor platform, (b) wireless gas sensor node, and (c) Arduino platform with connected actuators (LED, lamp, fan).

drops below a threshold, fan turns on when the temperature increases, and red LED starts to blink when potential fire is detected by the framework. We would like to note that the gas sensor node has a buzzer on board. This buzzer serves as an actuator (alarm) when the framework detects a potentially dangerous situation.

Arduino is connected directly to the internet using Ethernet. The next sub-section provides a description on how the devices and framework are interconnected.

3) Communication: The internet provides the sensors, actuators and framework with the communication infrastructure. The sensor nodes are connected to it through the gateway (laptop), while the framework and Arduino with the actuators are connected directly. In order to enable the framework cognitive mechanisms to interact with the Cosm (former Pachube) platform [24] is utilized. This platform allows the online, real-time submission, storage and access of sensor measurements. In this respect sensing and actuation feeds have been created. The sensing feed registers the measured values from the temperature, humidity and gas sensors. The cognitive mechanisms of the framework access this data through the Cosm platform and perform an analysis to determined the behavior of the CVO and consequently of the VOs and corresponding sensors/actuators. In case an action is deemed necessary as result of this analysis simple binary commands are submitted through the actuation Cosm feed. Arduino with three actuators on board and the gas sensor node with the buzzer read the commands and react on the event by turning the actuators on/off.

B. *Virtualization of Real World Objects*

As already mentioned a VO is the virtual representation of a RWO. The VO comprises of two parts; (a) the semantically enriched information for the description of the VO and (b) a software agent, developed as a RESTful Web Service (WS) [25], that implements a set of functions (e.g.: retrieval of measurements on environmental conditions, etc) related to the features/capabilities of the RWO.

The Resource Description Framework (RDF) [26], W3C standard model for data interchange on the Web, has been used for the representation and storing of the VO information in machine readable/understandable data. The VO information is stored as RDF Triples [25] in the form Subject – Predicate – Object, in the VO registry and constitutes the description of VO.

The RWOs offer functions that should be accessible through the VOs to other entities (VOs and CVOs). For this reason the virtual representation comprises the software entity part that as already mentioned is implemented as a RESTful WS, which is accessible via the Internet through a specific URI. Essentially, a VO is a Web Resource that is accessible through HTTP methods and offers services for the access to (use of) RWO functions. The use of REST WS enables the interoperability between various heterogeneous software and hardware technologies.

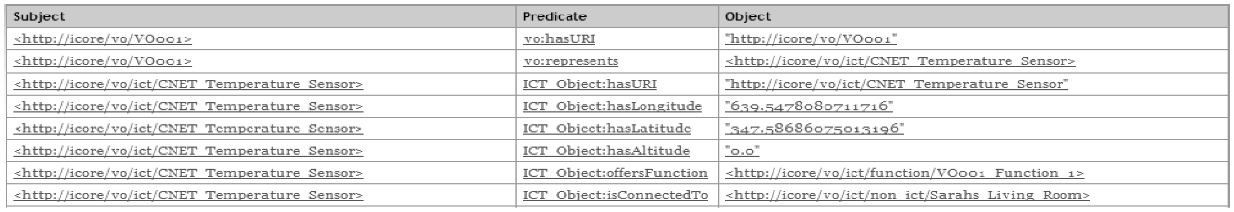
C. *Implemetation of VO and CVO registries*

Distributed registries in the system are used for the storing of information on VOs and CVOs. VO and CVO registries have been implemented as RDF graph databases with the use of Sesame repository [27], an extensible Java framework that supports the management of RDF data. The Sesame repository hosts the RDF graph databases and fully supports the SPARQL query language [28] for the execution of queries on the stored data. The entities that are available in the framework can communicate with the registries through the invocation of RESTful WS that are developed in the VO and CVO registry domains.

D. *Cognitive Mechanisms*

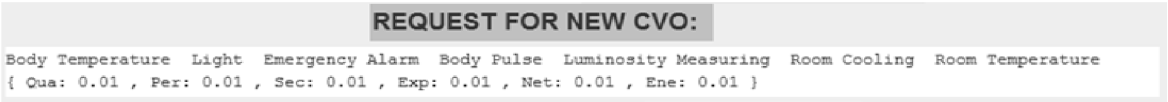
1) Request and Situation Matching: is triggered after the situation acquisition phase, when a set of extracted parameters is provided as input. The similarity metric-based matching algorithm is used to search and find an adequate match (i.e. existing CVO) in the CVO registry that can satisfy the current request and situation.

Communication between the entities is performed over HTTP in a RESTful manner. The underlying mechanisms, as well as the HTTP interfaces were implemented in Java, utilizing Jersey for the web services [29].



Subject	Predicate	Object
<http://icore/vo/VO001>	vo:hasURI	"http://icore/vo/VO001"
<http://icore/vo/VO001>	vo:represents	<http://icore/vo/ict/CNET_Temperature_Sensor>
<http://icore/vo/ict/CNET_Temperature_Sensor>	ICT_Object:hasURI	"http://icore/vo/ict/CNET_Temperature_Sensor"
<http://icore/vo/ict/CNET_Temperature_Sensor>	ICT_Object:hasLongitude	"639.5478080711716"
<http://icore/vo/ict/CNET_Temperature_Sensor>	ICT_Object:hasLatitude	"347.58686075013196"
<http://icore/vo/ict/CNET_Temperature_Sensor>	ICT_Object:hasAltitude	"0.0"
<http://icore/vo/ict/CNET_Temperature_Sensor>	ICT_Object:offersFunction	<http://icore/vo/ict/function/VO001_Function_1>
<http://icore/vo/ict/CNET_Temperature_Sensor>	ICT_Object:isConnectedTo	<http://icore/vo/ict/non_ict/Sarahs_Living_Room>

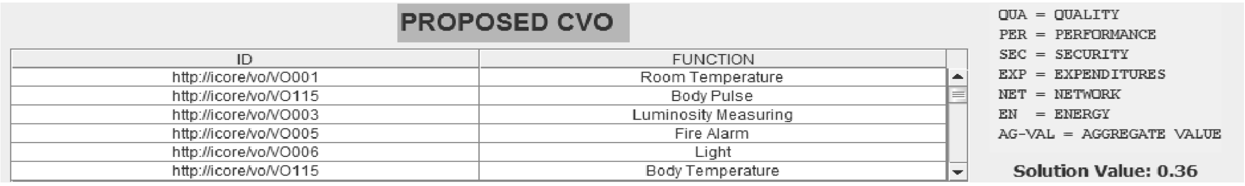
Figure 4: VO information stored as RDF triplets the Sesame repository



REQUEST FOR NEW CVO:

```
Body Temperature  Light  Emergency Alarm  Body Pulse  Luminosity Measuring  Room Cooling  Room Temperature
{ Qua: 0.01 , Per: 0.01 , Sec: 0.01 , Exp: 0.01 , Net: 0.01 , Ene: 0.01 }
```

Figure 5: Decision Making inputs - Request parameters and Policies for the new CVO.



PROPOSED CVO

ID	FUNCTION
http://icore/vo/VO001	Room Temperature
http://icore/vo/VO115	Body Pulse
http://icore/vo/VO003	Luminosity Measuring
http://icore/vo/VO005	Fire Alarm
http://icore/vo/VO006	Light
http://icore/vo/VO115	Body Temperature

QUA = QUALITY
PER = PERFORMANCE
SEC = SECURITY
EXP = EXPENDITURES
NET = NETWORK
EN = ENERGY
AG-VAL = AGGREGATE VALUE

Solution Value: 0.36

Figure 6: Decision Making outputs – A set of proposed VOs for the composition of CVO.

2) Decision Making: As described in sub-section III.B.4) the role of this component is to select the most appropriate combination of VOs. The Decision Making entity, using appropriate RESTful WS can interact with the "Request and Situation Matching", the VO registry and the CVO registry. Inputs of the Decision Making are presented through a graphical user interface (Figure 5). ILOG CPLEX [30] was used in the implementation of the decision making process. The ILOG CPLEX executes the objective function that takes into account the requested functions and policies and returns as a result the optimal composition of VOs (Figure 6), that fulfils the requested application.

E. CVO Implementation

As already mentioned (section III.B.1), the CVO is the derived solution that is produced by framework's cognitive mechanisms and comprising of VOs with different functionalities. The CVO after its creation is registered in the CVO registry and is activated by Decision Making mechanisms. The CVO uses eight different VOs for the monitoring of Sarah's health and climate conditions in her home. It exploits the VOs through the invocation of the RESTful WS that enable it to be able to interact with the Cosm online databases so as to read/write values from/to them. The sensors in Sarah's home perform measurements and store the measured data in the Cosm DBs. The CVO sends a request to Cosm so as to acquire the values of the measurements. After taking the measured data, starts the elaboration of them in order to create appropriate commands for the actuators. After the creation of commands, it uploads the commands values to Cosm DBs and the gateway, where the actuators are connected, performs a request for the acquisition of commands values. At this point should be highlighted that the decision making process for the creation of commands is based on service logic that is created as a result of the application requirements

VI. Conclusions and Future Work

This paper has presented a framework for the cognitive management of objects and applications in the IoT. The concept of VOs has been introduced with the aim of bridging the virtual world with the physical world. In addition the concept of mash-ups of semantically interoperable VOs has been introduced, namely CVOs enabling the dynamic creation of smart applications that can be *reused* outside of the context and the domain for which they were originally developed. An indicative smart home scenario was adopted in order to implement and validate the proposed framework.

The planning for future work involves the further implementation of functional entities that are designed but not yet implemented. Cognitive mechanisms for the management of the information regarding the users as well as mechanisms for the translation of application requirements will be further developed in order to enhance the system's functionality. In addition, it is planned to apply the cognitive management framework in further application scenarios, such as smart vehicular communications and smart city.

Acknowledgment

This article describes work undertaken in the context of the iCore project, 'Internet Connected Objects for Reconfigurable Ecosystems' (http://www.iot-icore.eu/). iCore is an EU Integrated Project funded within the European 7th Framework Programme, contract number: 287708. The contents of this publication are the sole responsibility of iCore project and can in no way be taken to reflect the views of the European Union.

Alexander Baranov would like to acknowledge the NATO Collaborative Linkage Grants CBP.CLG.984158 and Russian Federal Program "Development of Electronic Components and Radio Electronics" Grant No. 01.426.11.0050.

References

[1] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From Today's Intranet of Things to a Future Internet of Things: A Wireless- and Mobility-Related View," Wireless Communications, vol. 17, no. 6, Dec 2010, pp. 44-51.[Online]. Available: ACM Digital Library, doi: 10.1109/MWC.2010.5675777. [Accessed: Oct. 2012].

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," Computer Networks, vol. 54, no. 15, Oct 2010, pp. 2787-2805. [Online]. Available: ACM Digital Library, doi: 10.1016/j.comnet.2010.05.010. [Accessed: Oct. 2012].

[3] D. Miorandi, S. Sicari, F, De Pellegrini, and I. Chlamtach, "Internet of Things: Vision, Applications and Research Challenges", Ad Hoc Networks, vol. 10, no. 7, Sept 2012, pp. 1497-1516. [Online]. Available: ACM Digital Library, doi: 10.1016/j.adhoc.2012.02.016. [Accessed: Oct. 2012].

[4] R. Weinstein, "RFID: a technical overview and its application to the enterprise," IT Professional , vol.7, no.3, May-June 2005, pp. 27- 33. [Online]. Available: IEEE Xplore Digital Library, doi: 10.1109/MITP.2005.69. [Accessed: Oct. 2012].

[5] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks", Proc. of the International Conference on Mobile Data Management (MDM 07). Washington, DC, USA, May 2007, pp. 198-205. [Online].

Available: ACM Digital Library, doi: 10.1109/MDM.2007.36. [Accessed: Oct. 2012].

[6] V. Stavroulaki, Y. Kritikou, and P. Demestichas, “Acquiring and Learning User Information in the Context of Cognitive Device Management”, Proc. of International Conference on Communications (ICC 09), 2009, pp. 1–5. [Online]. Available: IEEE Xplore Digital Library, doi:10.1109/ICCW.2009.5208053. [Accessed: Oct. 2012].

[7] O. Vernesan, P. Friess, G. Woysch, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, M. Eisenhauer, K. Moessner, “Europe’s IoT Strategic Research Agenda 2012”, “The Internet of Things 2012: New Horizons”, in IERC - Internet of Things European Research Cluster , ISBN 978-0-9553707-9-3, pp. 22-23, Halifax, UK, 2012.

[8] D. Guinard, V. Trifa, F. Mattern, and E. Wilde, “From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices,” in: Dieter Uckelmann, Mark Harrison, Florian Michahelles (Eds.): Architecting the Internet of Things. Springer, ISBN 978-3-642-19156-5, pp. 97-129, New York Dordrecht Heidelberg London, 2011.

[9] M. C. Domingo, “An Overview og the Internet of Things for People with Disabilities,” Network and Computer Applications, vol. 35, no. 2, March 2012, pp. 584-596. [Online]. Available: ACM Digital Library, doi: 10.1016/j.jnca.2011.10.015. [Accessed: Oct. 2012].

[10] S. Duquennoy, G. Grimaud, J.-J. Vandewalle, “The Web of Things: Interconnecting Devices with High Usability and Performance,” Proc. of Embedded Software and Systems (ICESS 09), HangZhou, Zhejiang, China, May 25-27 May, 2009, pp. 323-330. [Online]. Available: IEEE Computer Society (CS) Digital Library, doi: 10.1109/ICESS.2009.13. [Accessed: Oct. 2012].

[11] CSA for Global RFID-related Activities and Standardisation (CASAGRAS2). Available: http://www.iot-casagras.org/ [Accessed: 13 Oct. 2012].

[12] FP7/ICT project SENSEI, Available: www.ictsensei.org, [Accessed: Oct. 2012].

[13] A.Somov, “Wildfire Safety with Wireless Sensor Networks”, Ambient Systems, vol. 11, no. 10-12(e4), Oct-Dec 2011, pp. 1-11. [Online]. Available: European Union Digital Library, doi: 10.4108/trans.amsys.2011.e4

[14] A. Somov, A. Baranov, A. Savkin, D. Spirjakin, A. Spirjakin, and R. Passerone, “Development of Wireless Sensor Network for Combustible Gas Monitoring”, Sensors and Actuators A: Physical, vol. 171, no. 2, Nov 2011, pp. 398-405. [Online]. Available: National Library of Australia, doi:10.1016/j.sna.2011.07.016, [Accessed: Oct. 2012].

[15] D. M. Doolin and N. Sitar, “Wireless Sensors for Wildfire Monitoring”, Proc. Conference on Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, San Diego, CA, USA, March 7–10, 2005, pp. 477–484.

[16] C. Lozano and O. Rodriguez, “Design of Forest Fire Early Detection System Using Wireless Sensor Networks”, Electronics and Electrical Engineering, vol. 3, no. 2, 2011, pp. 402–405.

[17] B. Son, Y.-S. Her, and J.-G. Kim, “A Design and Implementation of Forest-Fires Surveillance System Based on Wireless Sensor Networks for South Korea Mountains”, Computer Science and Network Security, vol. 6, no. 9, Sept 2006, pp. 124–130.

[18] A. Somov, D. Spirjakin, M. Ivanov, I. Khromushin, R. Passerone, A. Baranov, and A. Savkin, “Combustible Gases and Early Fire Detection: An Autonomous System for Wireless Sensor Networks”, Proc. of the First International Conference on Energy-Efficient Computing and Networking (e-Energy 10), Passau, Germany, April 13–15, 2010, pp. 85–93.

[19] W3C Naming and Addressing: URIs, URLs Available: http://www.w3.org/Addressing, [Accessed: Oct. 2012].

[20] D. H. Stefanov, Z. Bien, W-Ch. Bang, “The Smart House for Older Persons and Persons with Physical Disabilities: Structure, Technology Arrangements, and Perspectives”, Neural Systems and Rehabilitation Engineering, vol. 12, no. 2, June 2004, pp. 228–250. [Online]. Available: IEEE Xplore, doi: 10.1109/TNSRE.2004.828423. [Accessed: Oct. 2012].

[21] Libelium, WaspMote sensor platform. Available: http://www.libelium.com/products/waspmote, [Accessed: Oct. 2012].

[22] Arduino platform, Available: http://arduino.cc. [Accessed: Oct. 2012].

[23] A. Somov, A. Baranov, A. Savkin, M. Ivanov, L. Calliari, R. Passerone, E. Karpov, and A. Suchkov, “Energy-aware Gas Sensing Using Wireless Sensor Networks”, Proc. of the 9th European Conference on Wireless Sensor Networks (EWSN 12), Trento, Italy, February 15-17, 2012, LNCS vol.7158, pp. 245-260.

[24] Cosm Connect your world. Available: https://cosm.com. [Accessed Oct. 2012].

[25] Alex Rodriguez, RESTful Web Services: The basics, Nov. 2008, Available:http://www.ibm.com/developerworks/webservices/library/ws-restful/. [Accessed: Oct. 2012].

[26] Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, February 2004, Available: http://www.w3.org/TR/rdf-concepts/. [Accessed: 13 Oct. 2012].

[27] openRDF.org, Available: http://www.openrdf.org/. [Accessed: Oct. 2012].

[28] SPARQL Query Language for RDF, W3C Recommendation, Jan. 2008. Available: http://www.w3.org/TR/rdf-sparql-query/. [Accessed: Oct. 2012].

[29] RESTful Web services with Jersey. Available: http://jersey.java.net. [Accessed: Oct. 2012].

[30] IBM ILOC CPLEX Optimizer. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/. [Accessed: Oct. 2012].