# A Cognitive Management Framework for Smart Objects and Applications in the Internet of Things

Dimitris Kelaidonis[1], Andrey Somov[2], George Poulios[1], Vassilis Foteinos[1], Vera Stavroulaki[1], Panagiotis Vlacheas[1], Panagiotis Demestichas [1]

[1]Univeristy of Piraeus, Department of Digital Systems,
Karaoli & Dimitriou Str. 80, 185 34 Piraeus, Greece
{dkelaid, gpoulios, vfotein, veras, panvlah, pdemest}@unipi.gr
[2] CREATE-NET
Via alla Cascata 56/D, 38123 Povo di Trento, Italy,
asomov@create-net.org

**Abstract.** This paper proposes a cognitive management framework for the Internet of Things (IoT). The framework includes three levels of functionality: virtual object (virtual representation of real object enriched with context information), composite virtual object (cognitive mash-up of semantically interoperable virtual objects), and user/stakeholder levels. Cognitive entities at all levels provide the means for self-management (configuration, healing, optimization, protection) and learning. The paper also presents the implementation of the proposed framework, comprising real sensors and actuators. The preliminary results of this work demonstrate high potential towards self-reconfigurable IoT.

**Keywords:** Internet of Things, Cognitive Management, (Composite) Virtual Object, Self-Management, Smart Object

## 1. Introduction

The "7 trillion devices for 7 billion people" paradigm [1] yields that the handling of the amount of objects that will be part of the Internet of Things (IoT) requires suitable architecture and technological foundations. Internet-connected sensors, actuators and other types of smart devices and objects need a suitable communication infrastructure.

The appearance of RFID [2] and Wireless Sensor Network (WSN) [3] technologies allowed the users to capture and identify Real World Objects (RWOs) automatically. Due to distributed and autonomous nature of WSNs, a user can capture a RWO everywhere at any time.

The next research problem, that appeared with the evolution of the IoT domain, was 'how a RWO can be represented in a virtual world?'. A *Virtual sensor* approach is described in [4] that hides the implementation details from the user, but at the same time represents and includes a number of real devices, e.g. sensor, mobile phone, actuator, video camera. A lot of approaches, such as [4], have focused on the virtualization of the real devices. These works, however, lack smart management functionality of virtualized RWO, which is essential to overcome the technological heterogeneity and complexity of pervasive networks, to enhance context-awareness, to provide high reliability for reliable service provision, to support energy-efficiency of services.

The research works in [6] and [7] describe their efforts in the enrichment of each single virtual object with augmented smart functions to communicate with people and other virtual objects and be aware of context, situation, and past actions. A similar approach has been proposed in [8] where smart objects, support users in making decisions and taking mature and responsible actions.

A distinctive feature of CASAGRAS approach consists in the real object identification rather than its representation [9]. More specifically, the CASAGRAS derives an ontology for automatic real object identification in physical layer. This is a key requirement in the framework for the interfacing between real and digital world. The identified and digitized objects can be grouped in the networks and are then transmitted to the information management system through a gateway layer to provide the functional platform for application and services.

Another research works are focused on various technical aspects (hardware, software, etc.) as well as on application domains. For example, the SOFIA project [10] attempts to create a semantic interoperability platform to make information in the physical world available for smart services - connecting the physical world with the information world. In [11] the authors investigated how ontologies, runtime task models, Belief-Desire-Intention (BDI) models, and the blackboard architectural pattern may be used to enable semantic interaction for pervasive computing. In the approach presented in [12] the authors used explicit semantics to design abstracted models of connections between devices in a smart home environment. An architectural framework with three different levels of abstraction is the core concept of the SENSEI project [13]. The digital world has been classified into three abstractions: (a) resources (representations of the instruments), (b) entities (representations of people, places, and things), and (c) resource users (representations of real users which interact with resources and entities).

In response to the requirement of overcoming technological heterogeneity this paper proposes a cognitive management framework which aims to provide the means to realize the principle that any real world object (RWO) and any digital object, which is available, accessible, observable or controllable, can have a virtual representation in the IoT, which is called Virtual Object (VO). As opposed to [6][7][8], basic VOs can be composed in a more sophisticated way by forming Composite VOs (CVOs), which provide services to high-level applications and end-users. Essentially, a CVO is a mash up of different types of VOs. A CVO may be considered as a smart object that has functionality, and can communicate with other smart objects and/or applications. Moreover, a CVO may be used by the users in order to interact with other available smart objects.

The rest of this paper is structured as follows. Section 2 introduces a reader to the proposed cognitive framework for IoT. Section 3 presents an indicative scenario of the operation of the cognitive management framework applied in a smart home (assisted living) use case. Section 4 provides a detailed description of the framework implementation. Finally, concluding remarks and description of future work are provided in Section 5.

## 2. Framework Overview and Key Definitions

There are two key issues that need to be addressed for the successful realization of the IoT namely: (a) the abstraction of the technological heterogeneity that derives from the vast amounts of heterogeneous objects, while enhancing reliability and energy efficiency of network services and infrastructures; (b) the consideration of the views of different users/stakeholders (owners of objects & communication means) for ensuring proper application provision, business integrity and, therefore, maximization of exploitation opportunities.

In this direction, the solution proposed in this paper is a cognitive management framework comprising three levels of functionality, reusable for various and diverse applications. The levels under consideration are: (a) Virtual Objects (VOs) level. VOs are virtual representations of devices or digital objects (e.g. sensors, actuators, smartphones, music players, etc) associated to everyday objects or people (e.g. a table, a room, an elderly, etc) that hide the underlying technological heterogeneity. (b) Composite virtual objects (CVOs) level. Composite Virtual Objects exploit the Virtual Objects. They are cognitive mash-ups of semantically interoperable VOs, delivering services in accordance with the user/stakeholder requirements. Essentially, the CVO is a smart object with cognitive capabilities. (c) User/stakeholder level. User/ stakeholder related objects will convey the respective requirements. Cognitive entities at all levels provide the means for self-management (configuration, healing, optimization, protection) and learning. In this respect, they are capable of perceiving and reasoning on their context (e.g., based on event filtering, pattern recognition, machine learning), and of conducting associated knowledge-based decision-making (through associated optimization algorithms and machine learning). In the next subsections, the key concepts and the cognitive mechanisms of the three levels of functionalities are described in more detail.

### 2.1 Virtual Objects Level

The main entities at the Virtual Object (VO) level are the VOs (representing various RWOs) and VO registries which store VO descriptions and provide information on available VOs.

#### 2.1.1 Virtual Object

A Virtual Object (VO) is the virtual representation of Real-World Objects (RWOs). RWOs may be either digital objects with Information and Communication Technologies (ICT) capabilities or non-ICT objects. ICT objects include objects such as sensors, actuators, smart phones, etc. ICT objects may have a physical location and may offer various functions such as environmental condition measurements, location of objects/person, monitoring of places for security reasons, etc. Non-ICT objects are tangible objects of the physical world that do not have any direct ICT capabilities such as furniture, a room, fruits, a person, etc. Non ICT objects can be implicitly represented in the virtual world through their association with one or more ICT objects, which in turn are represented through VOs.

#### 2.1.2 VO Registry

The VO registry includes information about VOs that are available in the system. The information describes the associations between the VOs with ICTs and non-ICTs as well as the related data with these. Each VO is identified by a Uniform Resource Identifier (URI) and contains information on the association with an ICT object. The ICTs are also identified by a specific URI, have a specific physical location that is described with the coordinates and offer one or more functions. A function description provides information on its inputs and outputs and is also identified by a URI. Furthermore, each function has a set of costs and a set of utilities that are considered by the decision making process when targeting any specific policy enforcement. Moreover, information about associations between ICT objects with non-ICTs is included in the VO registry whilst the description of a non-ICT comprises of a URI and a non-ICT name.

### 2.2 Composite Virtual Object Level

The CVO level contains CVOs, the CVO registry, which contains information on the available CVOs, the Request and Situation Matching, and the Decision Making mechanisms.

#### 2.2.1 Composite Virtual Objects

CVOs exploit the functions provided by VOs. A CVO is a cognitive mash-up of semantically interoperable VOs that renders services in accordance with user/stakeholder perspectives and application requirements. CVOs are self-managed, self-configurable components, which exploit cognitive mechanisms to enable the mash-up and re-use of existing VOs and CVOs by various applications, also outside the context and domain for which they were originally developed. CVOs are smart objects created dynamically in an autonomous manner taking into account requirements deriving from the user, application and objects levels.

#### 2.2.2 CVO Registry

The CVO registry contains all information regarding the components of CVOs, how they are interconnected and the situation as well as the actual request parameters under which they were created or instantiated. In addition, it holds any explicit or implicit user feedback that is associated with the CVO lifecycle. Such records are exploited in forthcoming requests to improve the performance of the CVO deployment phase as well as the service quality for either the particular stakeholder or others requesting similar services.

### 2.2.3 Request and Situation Matching

The goal of the Request and Situation Matching functional entity is to identify past application requests that match closely enough to the present, incoming ones and the situations under which they were issued, so that the task of VO composition from scratch can be avoided under certain circumstances. In order to compare between past and present situations and requests, parameters that describe them have been identified. Request parameters consist of the set of functions requested and the costs or utilities specified to be maximized or minimized. Situation parameters consist of the time of day the request occurred, the area of interest, and the available VOs at that time. In order to enhance the filtering process, the requested functions can be matched with approximate ones, e.g. a video capture function can satisfy a requirement for an image capture function. The result of this mechanism is performance gain, as the process of creating a CVO from scratch is more complex. Moreover, reusability of resources is introduced in the system.

### 2.2.4 Decision Making

The Decision Making entity is triggered by the Request and Situation Matching mechanisms, interacts with the VO registry and its objective is to find the optimal composition of VOs that fulfils the requested functions and policies. It receives as input a set of available VOs, a set of functions for each VO, a set of utility and cost values for these VOs and finally weights for these utilities and costs. A correlation matrix provides the suitability level of an offered function of a VO with the requested ones. The result of the decision making process is the creation and activation of a new CVO. The description of the newly instantiated CVO is recorded in the CVO registry in order to be available for future requests.

## 2.3 User / Stakeholder Level

Users interact with the system via interfaces, and may request, configure and use applications or consume services. In order to enhance, among others, user experience, the system maintains user profiles that hold a collection of parameters regarding their preferences. Their feedback is used to build knowledge so that future preference estimations can potentially be derived. Further description of the user/stakeholder level mechanisms that the specification of the proposed framework designates follows in the next paragraphs.

### 2.3.1 Application Translation

Users provide a set of application requirements and policies for the preferred application while different policies may govern a specific domain. The Application Translation mechanisms map these requirements to service logic requirements and combine them with the user-requested as well as the local policies, if present. In turn, a set of request parameters is extracted and forwarded to the Situation Acquisition mechanisms.

### 2.3.2 Situation Acquisition

The Situation Acquisition mechanisms receive a set of request parameters from the application translation mechanisms. These mechanisms enable the cognitive management framework to be able to aggregate and infer relevant situational dimensions (e.g. time, place, VO states), in order to describe the current situation and anticipate changes to it. The output result of the situation acquisition mechanisms is a set of request and situation parameters that are provided to the Situation and Request Matching mechanism in order to be processed.

## 3. A Smart Home Use Case

This section presents an indicative scenario for the operation of the proposed framework. The scenario comprises two different (business) domains; (a) a smart home, where an elderly woman (Sarah) who has opted for an assisted living service lives in, and (b) a medical center, where doctors monitor Sarah's environmental conditions and health status remotely, using the smart objects that exist in smart home.

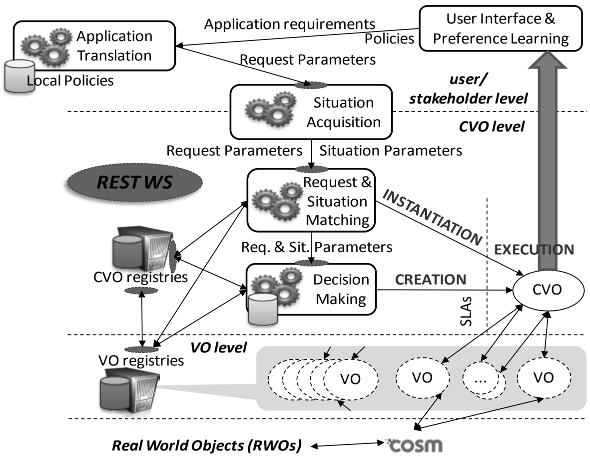


**Figure 1: An indicative operation scenario**

Firstly, a doctor through an appropriate user interface provides a set of application requirements. From the Application Translation and Situation Acquisition entities, a set of request and situation parameters are extracted to be forwarded to the Request and Situation Matching mechanisms, which, in turn, search in the CVO registry for a previously created CVO that could fulfil the requested application. If an appropriate CVO is not found, the provided parameters are sent to the Decision Making entity, which will select the most appropriate VOs to satisfy the requirements and policies in the best possible way, and will trigger the creation of a new CVO. The newly created

CVO, is registered in the CVO registry together with the situation parameters under which it was requested for future reference by the Request and Situation Matching entity. Finally, the doctor, or member of the medical staff, can use the dynamically created CVO to monitor the medical status of Sarah, Feedback regarding the operation of the CVO can be provided.

## 4. Framework Implementation

This section describes a first implementation of the proposed Cognitive Management Framework.

### 4.1 Real World Objects

The RWOs that have been used in the framework are a set of sensors and actuators. More specifically, the Waspmote platform by Libelium [14] is exploited that hosts three different sensors (temperature, humidity and luminosity sensors) for sensing of the environmental conditions in the Sarah's Smart Home. Three different actuators are used (a fan, a colored lamp and a LED) that are connected with the Arduino platform [15]. The sensors and actuators are connected to the Cosm (former Pachube) IoT platform [16]. Sensing and actuation streams have been created on Cosm to save sensing values and commands for the actuators. The sensors are connected to Waspmote which sends the sensors' measurements to Cosm's databases through a gateway. The instantiated CVO communicates with Cosm, gets the measurements from the sensing stream, generates commands for the actuators, according to the service logic corresponding to the requirements provided at the user/stakeholder level, and sends them to Cosm's actuation stream. The Arduino platform automatically polls the actuation stream. As soon as it receives the corresponding command(s) it (de)activates the respective actuator(s). The RWOs represented as VOs and the information that describing these is stored in the VO Registry.

### 4.2 VO and CVO Registry

The VOs/CVOs registries have been implemented as Resource Description Framework (RDF) [17] graph databases with the use of Sesame, an extensible Java framework for the management of RDF data that is provided by openRDF [18]. A Sesame repository can host RDF graph databases and supports the use of the SPARQL query language [19]. The use of SPARQL queries allows the storing, querying and inferencing for RDF data that exist in the system registries. Access to the VO/CVO registry is provided via RESTful Web Services [20].

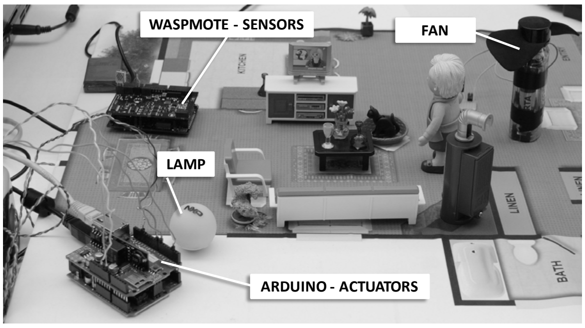


**Figure 2: View of RWOs in Cognitive Management Framework Implementation**

### 4.3 Request and Situation Matching

Application requests are issued through the graphical user interface (Figure 3) and then translated to machine-readable format and forwarded to the request and situation matching component via RESTful WS.

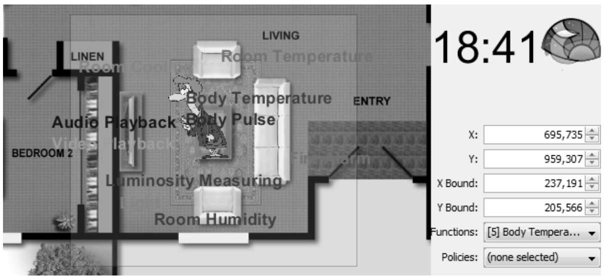


**Figure 3: Application request Graphical User Interface; VOs and the functions they can provide are shown**

At this point the application request can be compared with past ones in the search for an adequate match. Past request records that contain CVO components (VOs) with functions that are unavailable in the current situation (either them or approximate ones) are filtered out, as they definitely cannot fulfil the application goals. The remaining records are ranked based on a satisfaction-rate similarity metric and the highest ranked one is tested against the similarity threshold. The satisfaction-rate depends on the amount of total requested functions being available as well as their correlations and it is implemented as a score (i.e. sum) of these correlations between the set of the requested and the required CVO functions. Besides the functions, the overall similarity metric considers also the rest of the situation and request parameters.

### 4.4 Decision Making

Situation and request parameters are forwarded to the Decision Making (DM) entity. Afterwards, complete records of the available VOs are requested from the VO registry. Inputs of the Decision Making are presented through a graphical user interface. ILOG CPLEX 12.2 [21] was used in the implementation of the decision making process. More specifically, an objective function that is created and inserted in ILOG takes into account the requested functions, the correlation of them with the offered functions of the proposed VOs, the various policies and the respective values of these VOs features.

### 4.5 Instantiated CVO – Smart Object

The CVO is the result of the elaboration of application requirements. It is created as a Smart Object that is comprised of different VOs and delivers a set of services in an efficient cognitive way. The CVO communicates with the RWOs that are available in Sarah's Smart Home, through the Cosm's data feeds. More specifically, the CVO gets the real-time sensor measurements through a sensing feed and elaborates these values in order to draw conclusions about Sarah's home environmental conditions and her health. After the measurements elaboration, the CVO makes a set of decisions regarding the manipulation of available actuators, based on the service logic created as a result of requirements provided at the user/stakeholder level.

## 5. Conclusions and Future Work

This paper has introduced the concepts of Virtual Object (VO) and Composite Virtual Object (CVO) as means for abstracting the heterogeneity and complexity of the IoT infrastructure and for enabling the dynamical creation of smart objects exploiting various devices, not bound to specific application domains. The main contribution of this paper within the framework is, however, the cognitive functionality of VO and CVO which ensures their self-management (configuration, healing, optimization, and protection) and learning. The proposed concepts and framework were implemented and tested in the context of an assisted living use case. The preliminary results demonstrate high potential of the proposed framework. Future work will involve the implementation of functional entities that are designed but not yet implemented. Cognitive mechanisms such as the "application translation" and "learning" mechanisms as well as mechanisms for the "user profile" information management will be further developed in order to enhance the system's functionality.

**Acknowledgement.** 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.

## References


1. M. Uusitalo, “Global Vision for the Future Wireless World from the WWRF”, Vehicular Technology Magazine, IEEE, vol.1, no.2, pp.4-8, 2006.
2. R. Weinstein, “RFID: a technical overview and its application to the enterprise,” IT Professional, vol.7, no.3, pp. 27- 33, May-June 2005.
3. C.-Y. Chong, S. P. Kumar, “Sensor Networks: Evolution, Opportunities, and Challenges,” Proceedings of the IEEE, vol. 91, no. 8, pp. 1247-1256, 2003.
4. K. Aberer, M. Hauswirth, and A. Salehi, “Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks,” In Proceedings of the International Conference on Mobile Data Management (MDM '07). Washington, DC, USA, pp. 198-205, 2007.
5. Weinstein, “RFID: a technical overview and its application to the enterprise”, IT Professional, vol.7, no.3, pp. 27- 33, May-June 2005.
6. M. Beigl, H.-W. Gellersen, and A. Schmidt, “MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects”, Computer Networks, vol. 35, no. 4, 2001, pp. 401–409.
7. F. Mattern, “From Smart Devices to Smart Everyday Objects”, Proc. Smart Objects Conf. (SOC 03), Springer, 2003, pp. 15–16.
8. N. Streitz et al., “Designing Smart Artifacts for Smart Environments”, Computer, vol. 38, no. 3, 2005, pp. 41–49.
9. FP7 project CASAGRAS, http://www.iot-casagras.org/.
10. SOFIA project Website, http://www.sofia-project.eu/.
11. Gerrit Niezen, Bram J.J. van der Vlist, Jun Hu and Loe M.G. Feijs, “From Events to Goals: Supporting Semantic Interaction in Smart Environments”, SISS2010, IEEE conference ISCC2010, Riccione, Italy, 22 June 2010.
12. Bram J.J. van der Vlist, Gerrit Niezen, Jun Hu, and Loe M.G. Feijs, “Design Semantics of Connections in a Smart Home Environment”, DeSForM-2010, Lucerne, Switzerland, November, 2010.
13. EU/FP7 project SENSEI (Integrating the Physical with the Digital World of the Network of the Future, 215923), January 2008 - December 2010, “The SENSEI Real World Internet Architecture”, Version 1.1, March, 2010.
14. Libelium, WaspMote sensor platform, http://www.libelium.com/products/waspmote.
15. Arduino platform, http://arduino.cc.
16. Cosm, https://cosm.com/
17. Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-concepts/, accessed June 2012.
18. openRDF.org, http://www.openrdf.org/, accessed June 2012.
19. SPARQL Query Language for RDF, W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/.
20. RESTful Web Services: The basics, http://www.ibm.com/developerworks/webservices/library/ws-restful/, IBM Developer works page REST, Alex Rodriguez, Nov. 2008.
21. IBM ILOC CPLEX Optimizer, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.