A Smart Control System Solution Based on Semantic Web and uID

Esa Viljamaa, Jussi Kiljander, Juha-Pekka Soininen, Arto Ylisaukko-oja
Technical research centre of Finland - VTT
Oulu, Finland
firstname.surname@vtt.fi

***Abstract*— In this paper, a novel method to extent the functionality of the control system using ubiquitous object identification (uID) technologies and Semantic Web is presented. The method enables dynamic adding of new objects and relationships to the system. As a proof of concept, a reference implementation of the system that utilizes a mobile phone, object identification system, sensors and Smart Environment for highly dynamic control system setup, is introduced. As a result, the introduced method expands the functionality of the control system and makes it more dynamic and easier to setup. Generally speaking, the method combines the present object identification technologies and Semantic Web for advanced Internet of Things (IoT) utilization.**

Keywords- Smart environment, M3, Semantic Web, uID, Ontology

I. INTRODUCTION

The current vision of IoT is to create an infrastructure for uniquely addressed interconnected objects whose information and capabilities can be accessed even from the other side of the world. In addition to the unique addressing, the objects in IoT should also be able to interact and behave in smart way to provide more value for the system than just a sum of them. [1][2]

In order to realize this vision various kinds of technologies from low-power computing platforms to innovative methods for end-user interaction needs to be developed. In this paper we present how two fundamental challenges related to enabling IoT can be solved by combining uID [3] and Semantic Web [4] technologies in a novel way. The first challenge is related to identifying real world objects and for finding information related to these objects. Second challenge is to enable autonomous smart objects to interact with each other meaningfully. The challenges are tried to address in this study by combining ubiquitous identification technology with method of sharing the semantics of the information without priori standardization of use case specific data models.

The scope of the practical work in this study was in enabling more valuable and versatile systems through the use of technology and information mash-up Another target was also to demonstrate usage possibilities of uID and Semantic Web technologies for the IoT.

The developed method enables novel way to create and modify a setup or a behavior of an application on the fly with new technology enabling new applications and their utilization possibilities. The significance of the system is on its openness, flexibility and simplicity.

The paper first discusses background of the topic and then presents used methods and technologies. Following approach and implementation sections describe the validation of the method and finally conclusion recapitulates the study.

II. BACKGROUND

A. Interoperability in IoT

The interoperability in IoT can be roughly divided into three levels: communication, service and information. From the traditional OSI model perspective the communication level covers the layers from L1 to L4 whereas the service and information levels can be though as L7 layer technologies (some functionality of semantic and service levels can also be modeled with L5 and L6 layers).

The communication level interoperability deals with challenges related to transmitting data from one device to another. In the past, the interoperability research has mainly focused on this level and because of this there is lot of mature technologies from physical to transport level available. These technologies include cellular radios, Wi-Fi, Bluetooth, ZigBee, 6LoWPAN [5] and TCP/IP protocol stack, just to name a few. Especially the 6LoWPAN protocol stack is a very promising technology because it creates the backbone for IoT by providing the IPv6 based Internet for resource restricted devices

In the service level the interoperability is related to discovering and interacting with various services that compose the IoT ecosystem. For service discovery there are many technologies available such as the Bluetooth's service discovery protocol (SDP), Zeroconf [6] and Service Locator Protocol (SLP) [7], for example. In addition many interoperability frameworks such as Universal Plug and Play (UPnP) and Device Profile for Web Services [8], for example, define their own methods for service discovery. SOAP [9] and REST [10] are the most common ways to provide interoperability for client-service interaction. The aforementioned UPnP and DPWS are examples of SOAP based technologies. Plain HTTP is, of course, the most common protocol for RESTful services and WWW the best example of a REST based system. From the IoT perspective an interesting RESTful protocol

 ISBN: 978-1-61208-171-7

is the Constrained Application Protocol (CoAP) [11]. CoAP is a specialized web transfer protocol for machine-to-machine applications in constrained devices and networks. Basically the goal of CoAP is to provide the same for IoT that HTTP provides for the WWW.

The objective of information level interoperability is to define a data format for the information so that different devices and applications can share the meaning of the information and are able to interact with each other meaningfully. Traditionally data format has been defined for each use case separately, instead of using semantic interfaces.

From the IoT perspective the use case specific standardization model is not a feasible solution because of two reasons. First, the standardization process is usually very time consuming and it would take very long time to standardize all possible use cases for the IoT. Second, because there is no common model for presenting the semantics of the information it would be difficult to develop smart applications that utilize information produced by various devices in a cross-domain manner.

Semantic Web is a concept for next generation WWW where the semantic interoperability issues are solved by utilizing ontology based model for presenting the meaning of information [4]. In ontology based model the semantics of information is modeled as classes and relations between those classes. The W3C's Semantic Web Activity has developed many technologies such as Resource Description Framework (RDF), RDF Schema (RDFS) [12], Web Ontology Language (OWL) [13] and SPARQL [14] to realize the ontology based interoperability in the WWW. In Smart Environment domain these Semantic Web ideas and technologies have been utilized in [15, 16] where a semantic interoperability solution called M3 has been developed [17].

B. M3 concept

M3 is a concept for utilizing the Semantic Web ideas and technologies to provide semantic level interoperability between devices in physical environments. By utilizing the ontology based information model the M3-based software agents can more autonomously interpret the meaning of information and therefore obtain greater degree of smartness and flexibility than could be achieved with traditional use case specific data models. M3 utilizes RDF, RDFS and OWL for presenting the semantics of information in a computer-interpretable manner. In the core of M3 is a functional architecture that specifies how the semantic information can be accessed in a physical space. The M3 functional architecture consists of Knowledge Processors (KP) and Semantic Information Brokers (SIB). SIBs are basically shared RDF-dabases of semantic information that provide publish/subscribe based interface for KPs. The role of KPs is to provide applications for end-users by interacting with each other via the SIB. Smart Space Access Protocol (SSAP) defines the rules for KP-SIB interaction. M3 utilizes existing solutions for the communication and service level meaning that it is possible to implement the SSAP protocol with different service and communication level technologies. Figure 1 shows the functional architecture of the M3 concept.

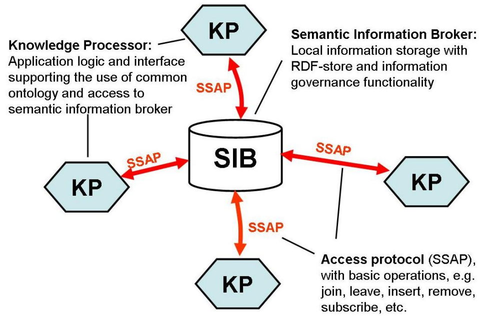

Figure 1. M3 functional architecture.

C. Object identification

In IoT concept, an object addressing and identification have been one of the most important matters. Since the IoT is considered to be world-wide, it is important that every entity can have its own unique identifier and the address space is large enough.

In object identification, there are two fundamental parts: an object identity reading method and a system providing a unique identity. At the moment there are couples of systems providing an identity to the object: uID and EPCglobal being probably the best known technologies.

EPCglobal defines electronic product code framework for example for a supply chain use. The developer research group of the EPCglobal is targeting to get EPCglobal to be the backbone of the global IoT infrastructure.

uID is a technology agnostic object identification system that provides 128 bit expandable address space for any kind of object identification. Due to its nature, it could be used through different kinds of tagging methods. uID system was developed by YPR laboratory in Tokio University. uCode is an identifier instance of the uID system. uID address sharing architecture is three tiered maintained by uID center in Japan. uID address subspaces have been allocated from uID center to top level server tier maintained by nonprofit organizations and further from top level servers to second level servers maintained by e.g., companies. Typical use case of uID is shown in Figure 2, where uCode is read by a mobile device and sent to the resolution server. A received IP address is then resolved in the information server for product or other relevant data. Resolved IP data can contain e.g., product data or tourist attraction information. In addition to recent uID architecture, a semantic resolution technology called uCode relation model for uCode is been developed to diversify the uID usage. [3]

 ISBN: 978-1-61208-171-7

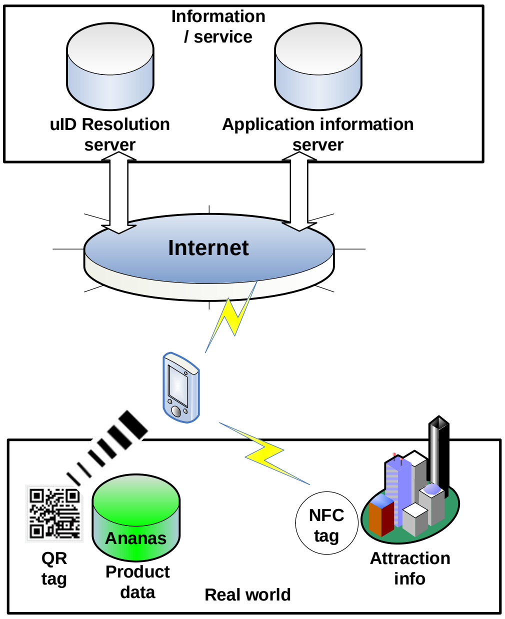

Figure 2. A typical uID utilization scheme.

For the object identity reading method almost any coding method can be used. The most common ways to read object identity are radio technologies RFID or NFC, optical tag s like QR-code or ordinary bar code or even sound coded tag.

III. APPROACH FOR IOT SOLUTION BASED ON SEMANTIC WEB AND UID TECHNOLOGIES

We ground our approach on the M3 -concept based Smart Environment. The functionality of the Smart Space is then extended by sharing uCodes and information associated to them. In addition to the unique addressing provided by uCode, simple, resource constrained objects in the Smart Space can be linked to modifiable data through the uCode resolving. With the aforementioned features, the system is reaching the IoT vision of the objects with unique addressing identifiers and smart interactions between the objects.

In the other words, the objective of this first phase integration is to improve the quality of service in local M3-based Smart Environment by utilizing information of uCode tagged objects. In this first phase we only address the following scenario:

1. uCode client reads the uCode
2. uCode client resolves the address of the information service that hosts information about the tagged object.
3. KP serializes the information to machine-interpretable format and publishes the information to a local SIB.
4. Other KPs utilize the information about the tagged object to improve the quality of services they provide for the end-user.

Figure 3 illustrates the system model for the IoT solution based on M3 and uID technologies.

In this paper the entity capable of utilizing the M3 and uID technologies is called a Smart Object. Smart Object contains the uCode client and KP entities. Typically the Smart Object is a software application for example in a mobile device.

Our approach to improve the M3-based smart environments with uID technologies is very user centric. The user can select the tagged objects that she wants to include to her personal smart environment by “touching” them with her Smart Device. When the uCode client retrieves the uCode from the tag it first contacts the uCode resolution server to obtain the address of the information service that hosts information about the tagged object. There are some suggestions how the reader application would know that it is an uCode in question. uID specification could be added directly to the NFC standards or the code could have a trailing string element “ucode:” An address of the used uCode server could be the highest in hierarchy i.e., uID center server or the server address could be hardcoded to the client application as in our case. After the uCode resolution is complete the client is able to fetch information about the object from the corresponding information service.

The information service can be basically any kind of server that contains some information about the tagged object. In the simplest case the information service is a web server that presents the information about the tag in a web page. More complex information services contain information about multiple uCodes. For these information services the uCode is passed as query parameter to indicate what information is requested. It is also possible to use RDF databases or even SIBs as a uID information service. In these cases the uCode Resolution server has many options for the response URIs. First option is that the URI can be only the SIB address and the Smart Object needs to specify the SPARQL query that requests the necessary information about the object. Second option is that the URI contains also the SPARQL query that request for example the rdfs:Class of the object. Table 1 illustrates these three types of information services, example URIs returned by the uCode resolution server and responses returned by the information services. In the URIs presented in the table the host part means the address of the information service. With HTTP this is either the hostname or IP address port pair.

 ISBN: 978-1-61208-171-7

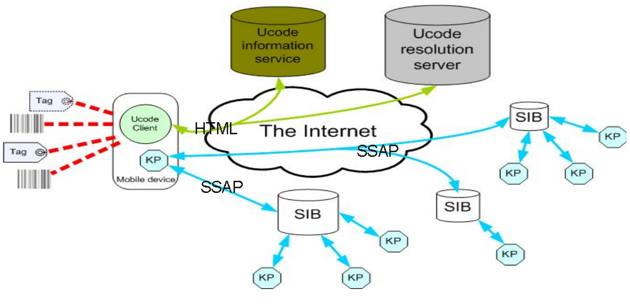

Figure 3. System model of open IoT with M3 and uID technologies.

TABLE 1. URIS AND RESPONSES FOR EXAMPLE INFORMATION SERVICES

Service type	URI returned by the uCode Resolution Server	Response
Web Server	http://host/objectInfo	Web page
Arbitary database	http://host/object?UCODE= 1A	Object presentation in arbitrary data format
SIB	ssap:// host	Depends on the query specified by the Smart Object
	ssap:// host /sparql?query= SELECT ?class WHERE{1A rdf:type ?class }	The URI of the RDFS/OWL class the object belongs to

In SSAP the host can also be the address of some lower level communication technology such as MAC address channel pair of Bluetooth or 802.15.4 radio. To obtain more compact presentation in the table the non-significant zeros are removed from the ucode 000000000000001A.

After the Smart Object has obtained information about the tagged object it publishes the information into the SIB to make the information accessible to devices in a local Smart Environment. Typically the data in the information service is not in RDF format and therefore the Smart Object needs first to serialize the information to common machine-interpretable data format e.g., RDF. This, of course, requires that the Smart Object is familiar with the application specific data model used by the information service. When the Smart Object publishes information about the object it uses scheme “ucode:” to inform other KPs that the resource identified by the URI is an uCode. After information about the tagged object has been published to the local smart environment the devices and applications are able utilize the information about the tagged object to improve the quality of their services. An example of this is presented in the section

IV. IMPLEMENTATION OF IOT BASED HOME GARDENING SYSTEM

In order to demonstrate the approach, an example application was implemented. The implementation for a plant moisture control system with help of a Semantic Web technologies and the uID was implemented. The implemented system features intelligent simple building blocks, information access from Internet for very simple objects and dynamic data relationship definition.

The plant moisture control system supervises plant moisture levels using wireless moisture sensors and announces low levels to the operator Smart Device. Using the M3- Smart Environment and uID object identity technology, mixed plant species preferring different moisture levels are paired with wireless moisture sensor modules in very flexible and useful way.

A. Smart Gardening System

The implementation contains following main components that can be seen in Figure 4.

1. A Google Nexus S Android smart phone acts as an operator user terminal. A terminal program with KP is run on the smart phone. The KP takes care of SIB insertions of flower pots, modifications on pot-sensor pairings and queries for unaccepted moisture values. Used interfaces are NFC reader, camera for optical tags and WLAN for SIB connections and uCode resolution and information server connections.
2. An operator presence NFC tag attached to a demo room is used by operator to join to be a one object in the Smart Space by touching the tag with the Smart Device.
3. There are three potted plants with an uCode printed on the pot using a QR coding. uCode is resolved by the user terminal and received application data with a flower minimum accepted moisture value among others is inserted to the SIB
4. For the moisture measurements there are three wireless Active Tag sensors that are based on Econotag hardware platform with a moisture sensor and an NFC tag. The Active Tag has a sensor ID on NFC tag and it runs a sensor program and a KP for data interchange. The KP communicates with SIB over IEEE 802.15.4 radio interface using it to insert their presence and update their moisture values to SIB

 ISBN: 978-1-61208-171-7

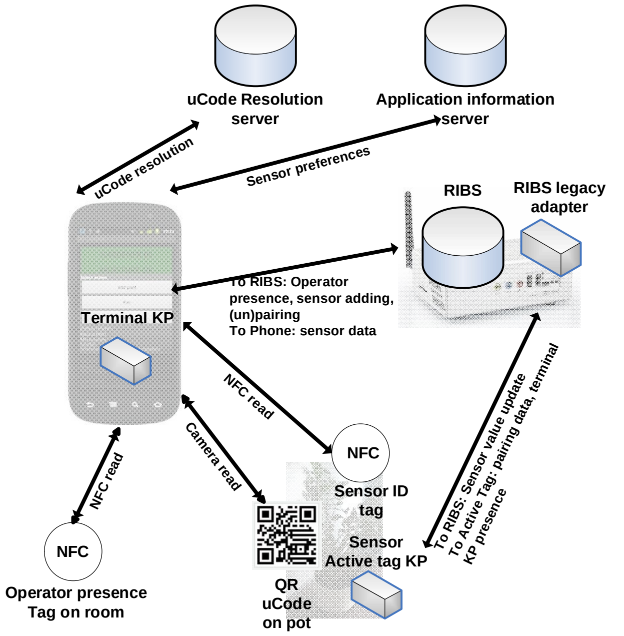

Figure 4. The implementation of uID and semantic web based IoT solution.

5. A Via Artigo A1100 Linux Ubuntu PC with SIB running on it is a core of the demo implementation. It has a 3G mobile internet connection shared over WLAN. The user terminal uses the WLAN and its 3G network bridge to communicate with SIB and uID and information servers in Internet. Moisture sensor KPs communicate with RIBS over IEEE 802.15.4 radio. To allow better support for low capacity devices we utilize a SIB implementation called RDF Information Base Solution (RIBS) and Word Aligned XML (WAX) version of the SSAP messages. The most notable difference between WAX and basic XML serialization of the SSAP is that in WAX each tag is just one word long (32 bits) and payload is word aligned. Because of this the SSAP/WAX format causes less overhead and is easier to parse than the basic SSAP/XML serialization. Using WAX it is easier to deploy embedded moisture sensors to run KP and therefore be able to be a part of the semantic system.
6. And finally an uCode resolution server and an information server are used to get flower minimum moisture and ID values.

B. *Ontology for Gardening Application*

In addition to the system model of the implementation the ontology model is very important part of M3-based solutions. Figure 5 presents the ontology model used by the gardening application.

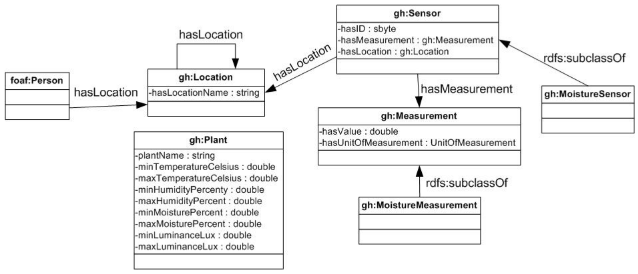

Figure 5. Gardening ontology.

The Sensor class presents the virtual counterpart of an entity able to measure its surroundings and present the measurement with a value and a specific unit of measurement. The Sensor class has properties specifying the location of the sensor and measurements made by the sensor. The Measurement class is used to present the information of the measurements made by the sensor. The Measurement class provides properties for presenting the value and unit type of the measurement.

The Plant class represents the physical plants. The information presented by the Plant class consists of plant name and preferences for environmental conditions such as temperature, luminance, soil moisture and humidity.

The Location class present the virtual counterpart of a physical location such as city, house, room or a pot, for example. The hasLocation property is used to associate resource with a Location class instance.

C. *Example use case scenario*

An example usage of the implementation is as follows.

An operator can announce his presence by touching the presence NFC tag with the phone in order to insert itself as an object to the SIB.

After the insertion the smart phone application queries constantly the SIB for sensor values and plant moisture values bound to the certain sensor if there's any. If the moisture value exceeds the reference, the phone alerts the operator showing the plant ID needing more water in the screen of the Smart Device.

By reading optically a QR tag with uCode on the pot, the operator can, by resolving the uCode and a web service bound to it, read plant's ID and minimum moisture reference data. Using the plant data, the operator can then either insert the plant ID with moisture reference to the SIB or pair/unpair earlier inserted plant ID with a desired moisture sensor ID by touching the sensor NFC tag with the Smart Device.

Meanwhile the Active Tag measures plant moisture values updates them to SIB and polls for the corresponding minimum preference moisture value and the operator presence from the SIB. If the moisture value is less than preference and the operator is present, an indication LED is blinked.

Interactions between different components of the implementation have been described in Figure 6.

 ISBN: 978-1-61208-171-7

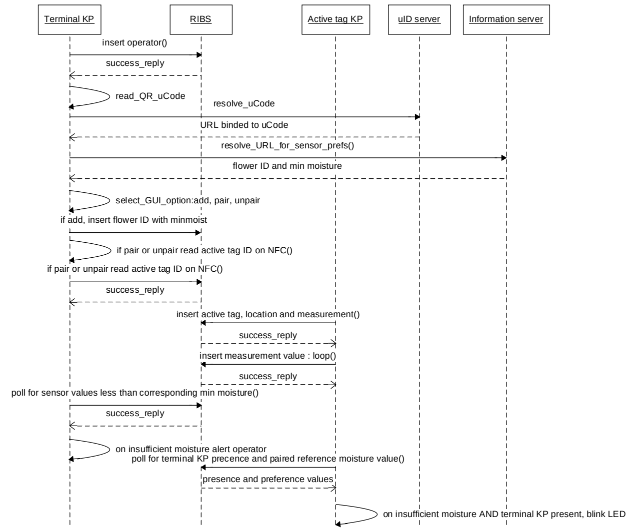

Figure 6. Interaction flowchart of the implementation.

V. CONCLUSIONS

In this paper a novel approach to use a general object identification system and Semantic Web for IoT to extent the functionality of the control application was presented. The idea of the approach was firstly to use M3-based interoperability framework to provide base functionality for a sensor – actuator network and secondly extent its functionality by using the uID-based object identification system for a reference data access. As a result the method for very easy-to-use and flexibly configured controlling scheme was achieved. The approach was demonstrated by implementing the home flower moisture control system on a smart phone, local semantic broker on PC and remote server environment. The implemented system worked as expected.

The developed system is very flexible and easy to use. The operator can easily pair a sensor to control the desired entity in the system like a certain flower in the implementation. The sensor and controlled entities do not need to know anything of each other since the binding is done through the semantic broker. The system has also been constituted using open components and technologies. Also the open-source semantic framework with open object identifier system with off-the-self components makes the system reachable for all potential users.

The system could be used in applications having a need to easily generate and modify dependencies or controlling schemes between entities using only cheap tag technology like NFC or QR.

In the future the similar systems with hundreds of nodes should be tested in order to test a scalability of the system. The broker should also be used from the remote location to maximize the potential application area. Also more complex ontologies and dependency scenarios between the system entities should be generated and tested to take a full potential out from the system. Speaking of new technologies for IoT the M3 with CoAP and 6LoWPAN technologies on the service and communication level could also provide interesting possibilities for novel applications.

ACKNOWLEDGMENT

This work was done in Open Ubiquitous Technologies (OPUTE) –project funded by TEKES (Finnish Funding Agency for Technology and Innovation). Also the contribution of the YPR laboratory from the University of Tokyo is highly appreciated.

REFERENCE

[1] M. Weiser. (2002, The computer for the 21st century. *Pervasive Computing, IEEE 99(1),* pp. 19-25.

[2] L. Tan and N. Wang. Future internet: The internet of things. Presented at Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on. pp. V5-376 - V5-380

[3] N. Koshizuka and K. Sakamura. (2010, Ubiquitous ID: Standards for ubiquitous computing and the internet of things. *Pervasive Computing, IEEE 9(4),* pp. 98-101.

[4] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *The Scientific American,* May 17. 2001, pp. 34–43.

[5] Z. Shelby and C. Bormann. (2010, *6LoWPAN: The Wireless Embedded Internet* Available: http://www.google.com/books?id=75JAV_4ATwgC.

[6] D. Steinberg and S. Cheshire. (2005, *Zero Configuration Networking: The Definitive Guide* .

[7] E. Guttman. (1999, Service location protocol: Automatic discovery of IP network services. *Internet Computing, IEEE 3(4*p. 71-80.

[8] Anonymous (2009, 1 July). Devices profile for web services (DPWS), specification. Last access *2011(10 June),* .

[9] Anonymous (2007, April 27). Latest SOAP version. *2011(June 10),* .

[10] R. T. Fielding and R. N. Taylor. (2002, May). Principled design of the modern web architecture. *ACM Trans.Internet Technol. 2(2),* pp. 115-150. Available: http://doi.acm.org/10.1145/514183.514185.

[11] Anonymous "Constrained Application Protocol (CoAP), draft-ietf-core-coap-06," vol. 2011, May 3, . Last access 2011 (June 10).

[12] [Anonymous (2004, 10 February). RDF vocabulary description language 1.0: RDF schema, W3C recommendation 10 february 2004. Last access *2011(10 June),* .

[13] D. L. McGuinness and F. van Harmelen. (2004, 10 February). OWL web ontology language overview. Last access *2011(10 June),* .

[14] Anonymous (2008, 15 January). SPARQL query language for RDF, W3C recommendation 15 january 2008. Last access *2011(10 June),* .

[15] Anonymous (2009, SOFIA project - smart objects for intelligent applications. Last access *2011(June 10),* .

[16] Anonymous (2011, June 6). DIEM project, devices and interoperability ecosystem. Last access *2011(June 10),* .

[17] P. Liuha, J. Soininen, and R. Otaolea. SOFIA: Opening embedded information for smart applications. Presented at Embedded World, 2010.

 ISBN: 978-1-61208-171-7