Open Multi-Technology Building Energy Management System

Jaime Caffarel, Guillermo del Campo-Jimenez, Jorge M. Perandones, César Gomez-Otero, Rocio Martínez and Asunción Santamaría

Abstract— **Energy Efficiency is one of the goals of the Smart Building initiatives. This paper presents an Open Energy Management System which consists of an ontology-based multi-technology platform and a wireless transducer network using 6LoWPAN communication technology. The system allows the integration of several building automation protocols and eases the development of different kind of services to make use of them. The system has been implemented and tested in the Energy Efficiency Research Facility at CeDInt-UPM.**

Keywords: Home Automation, Ontology, OSGi, RDF, SPARQL, Wireless Transducer Network, 6LoWPAN, Smart Meter, Energy Efficiency in Buildings.

I. INTRODUCTION

Energy consumption in buildings, accounts for between 20% and 40% of the overall amount of energy used in developed countries [1]. The integration through a common infrastructure of different building systems such as lighting, HVAC (heating, ventilation and air conditioning), security or life safety is important to provide intelligent management services with the aim of improving energy efficiency. Moreover, gathering information related to the use of energy made by devices, users and systems within a building is useful to improve the efficiency of electricity distribution systems, since it provides accurate information to avoid peaks and dips in energy demand.

Building Energy Management Systems (BEMS) aim to reduce energy consumption by monitoring data collected from sensors and controlling electrical devices and systems such as lighting or HVAC. Nevertheless, available commercial systems present several issues. On the one hand, interconnection among devices that use different communication technologies is a complex and expensive task and it usually entails a reduction of functionalities compared to those available systems that support just a single technology [2]. On the other hand, no additional external services can be added to these systems, limiting the development of future functionalities.

Real time metering systems of electric line parameters (such as current, voltage or real power) provide accurate information related to building electrical consumption. This information is essential to improve electricity generation and distribution in a smart grid. Most of the distribution board metering devices proposed so far only measure electric current and consequently the power consumption is calculated assuming a constant voltage from the electric line[1] [3]. Other devices measure the line voltage as well but just on one electric line[2]. Commercially available meters which fulfill these requisite are too expensive for home-oriented services.

In order to solve these issues and encourage smart cities development we propose an Energy Management System for buildings, designed to ease the implementation of new services and the integration of existing control systems within the building. The system consists of a Management Platform (Bat-MP) that enables the integration of different building control technologies, independently of the automation protocol used. Supported protocols are: LonWorks, KNX, ModBus, EnOcean and X10, although the Bat-MP could also be expanded to support other ones. The system also implements 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks), an open wireless communication standard especially designed for low-power devices. Bat-MP uses an ontology-based model to unify in a single data structure all the information related to building systems, devices and parameters measured, allowing data sharing and offering an open platform for the development of new services.

In addition to the management platform, the system also includes a Wireless Transducer Network (BatNet) which allows real time metering of electric line, ambient parameters, electrical loads and control of devices. A smart meter device which measures current and voltage, capable of monitoring up to seven lines within the same phase is presented. The consumption data measured by these devices is transmitted to the Bat-MP and stored in the ontology model, allowing the further sharing of the data.

Section II presents a general description of the building management system. In section III Bat-MP system architecture and its three main layers are detailed. Section IV describes the WTN devices focusing on the smart meter.

Finally, conclusions are presented in Section V.

II. SYSTEM DESCRIPTION

The Building Energy Management System consists of a Management Platform (Bat-MP) which controls sensors and

[1] Synetica Wi-CT – EnOcean® Wireless Meter: www.synetica.net.

[2] Orbis CONTAX 2511 SO: www.orbis.es.

actuators from different building management technologies, including those from the BatNet.

The BatNet is connected to the Bat-MP through one of its nodes which acts as coordinator. The Bat-MP collects data from sensors (temperature, illumination, humidity or presence) and sends control orders to the actuators (HVAC, lamps or blinds). Bat-MP connection to Internet enables the compilation of useful external information (weather forecast, user location, etc.) as well as the communication with the utility to provide metering information or to allow the implementation of demand response services (Fig. 1).

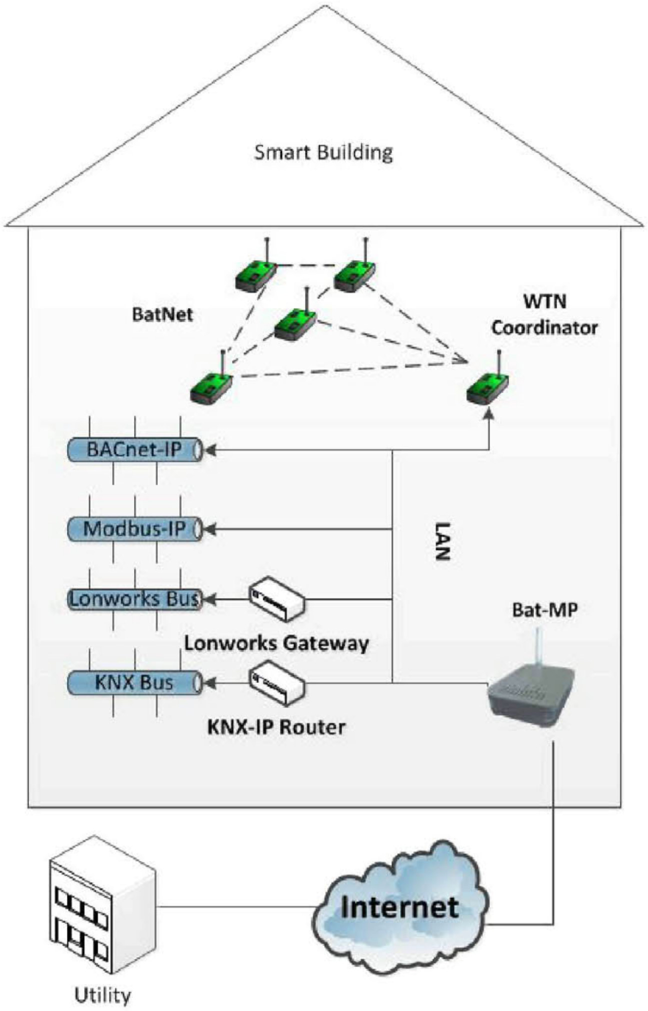

Figure 1: Structure of the Wireless System for Energy Efficiency in Buildings.

This system architecture not only eases the building energy management but also allows the integration of additional services such as safety (fire and flood detection), surveillance (video and access control) and multimedia (intelligent video and music). Besides, it can be used to manage external systems within the Smart City as gardening, street lighting or electric car battery recharging.

III. BAT-MP (BUILDING MANAGEMENT PLATFORM)

The Bat-MP consists of a Java-based management platform connected to the different control technologies inside a building through a local area network.

The Bat-MP has been designed as a middleware aimed to support different kinds of services which can thereby manage all the building systems using home automation protocols.

The main purpose of the middleware is to provide resources and tools to develop services of general interest in every building. This approach requires the middleware to solve the following issues:

- Allow interaction with the different network protocols used in building automation, such as KNX, LONWorks, BACNet, or ModBus, in a unified way [4].
- Provide a model to represent heterogeneous information of the building such as physical quantities, devices location, control information or user actions and the relationships among all of them [5].
- Allow third-parties to develop new services which make use of the platform in order to manage the building systems, needless to know low-level technological aspects.

To meet these requirements, the system has been designed in three layers (Fig. 2):

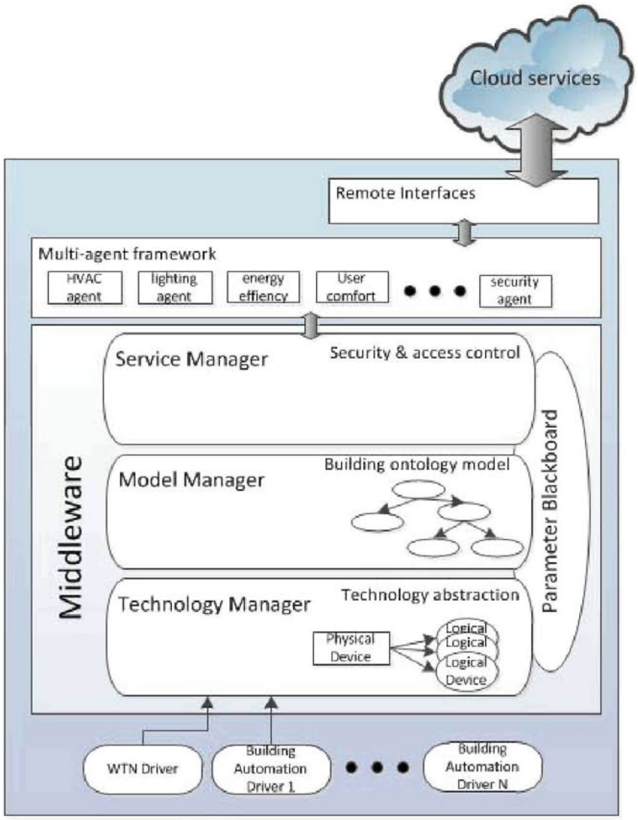

Figure 2: Platform software architecture.

a) Technology Manager: The main goal of this layer is to ease the integration of different building technologies and protocols, as well as other information sources (for example, climate data web services providing weather forecast in the geographical location of the building). This objective is achieved by the use of two concepts. The first one, Physical Device, defines each device or system using its control specific characteristics. The second one, Logical Device, is used to group and control different devices by their common functionalities or physical location. For instance, the set of lights which are located closer to a window can be controlled altogether independently of the rest of the lights within the room, by creating a Logical Device for them. Logical and Physical Devices are associated to one or more Parameters. Each one of these Parameters represents a physical quantity or state (such as temperature, brightness level or on/off state) which can be directly modified by these devices.

Thus, Technology Manager provides an abstraction layer aiming the direct interaction with the physical quantities, since the control actions are described in terms of operations performed directly over the Parameter, not as actions performed over specific devices. For instance, a Parameter "Temperature" includes a set of functions to increase or decrease the temperature of a certain room or area. These functions are common to every Parameter "Temperature" within the building, independently of the HVAC systems installed. Parameters are shared with the upper layers through a common resource (parameter blackboard). Thereby upper layers (Service and the Model Manager) are allowed to interact with the devices and their functionalities transparently and independently of the underlying technology i.e. hiding the implementation details.

In order to test the platform in a real environment, a set of control drivers has been integrated in the Technology Manager. Provided that the Bat-MP has physical connection to a KNX bus, the Calimero-KNXnet/IP [6] library allows the platform to communicate directly with KNX devices, without needing a KNX-IP gateway. LonWorks, ModBus, EnOcean and X10 drivers over IP have been implemented, letting the platform interact with existing facilities by means of gateways between IP and the specific bus of each technology.

b) Model Manager: The amount of information that an intelligent building should manage includes the following:

- Structural characteristics such as the number of floors, number and distribution of rooms or geographical orientation (in the northern hemisphere, rooms facing south are generally warmer than those facing north).
- Information about the electric systems of the building, such as sensor networks, security devices and lighting or HVAC systems.
- Physical quantities which can be either measured or modified (or both), such as temperature, light level, electrical consumption or gas concentration levels, like carbon monoxide.
- Information related to the use people make of the building, such as user timetable, specific usage of every area or user profiles.

The use of ontologies to represent this knowledge is becoming increasingly widespread [7]. By using an ontology-based system, all the building data described above are stored in a structured way and the relationships between all of them are explicitly described. Furthermore, the use of ontologies allows publication, sharing and extension of the data model. The Model Manager uses a Resource Description Framework (RDF) [8] building data model to describe all the entities, characteristics and properties of a building. RDF is a standard model for data interchange on the Internet which uses statements in the form of subject-predicate-object expressions. In order to enhance the system sharing capabilities, a SPARQL Protocol for RDF [13] interface is currently under development.

An ontology-based model of the building has been developed in order to store all the information described above. The main concepts of the model are:

- *Room*. Every building contains at least one room, with its own characteristics, such as size or geographical orientation (or in case the room has windows, the direction it is facing). In order to simplify the modeling process of the building, the location of every room within it is only defined by the adjacent rooms and their orientation.
- *Device*. Every room can contain zero or more devices that can be either a sensor (which gives a measurement of a physical quantity) or an actuator (which can perform actions in order to change certain magnitudes inside a room, such as a light dimmer).
- *Parameter*. A physical magnitude is represented as an object that contains information about its current value, the available actions over it and the units in which it is measured. Each parameter has, among others, "Measurement" attribute which stores the timestamp and the value of each measured physical quantity. The set of actions that can be performed over a parameter is determined by the devices present in the room. For instance, if a room contains a light installation that, due to its control technology, can just be switched on and off but not dimmed, only those two actions will be available in the parameter.

This ontology describes sensors and observations and all its associated concepts, since the main goal of the Model Manager is to represent and work with parameters in a simple way.

The Model Manager manages queries written in an RDF query language, validates them and retrieves the information from the database, allowing thereby the sharing of building data among different systems. The Model Manager uses the RDF Query Language SPARQL in order to manipulate the data stored in the database, considering that it is a W3C (World Wide Web Consortium) Recommendation [9] when implementing ontology databases.

c) Service Manager: The main purpose of this layer is to provide the different services a simple and uniform way to access to the platform, allowing them to work with the abstract model of the building without having to know low-level technological details. For instance, if the purpose of a climate service is to control the temperature of a certain room or area, the service should only take into account physical quantities which are related to climate, independently of the control technologies of the devices.

With the aim of allowing the development of remote services, the Service Manager also provides a common interface to access to the Management Platform through the Internet. Thus, a web service has been implemented using a service-oriented architecture (SOA). This interaction model is based on a message exchange process between the entities involved which allows a loosely coupled interaction between the services and the platform. The main goal of this design model is to achieve an easier integration of different services, using defined protocols to communicate with them. This approach gives the developer liberty to design services using different technologies or programming languages, since the only thing needed is the format of the messages exchanged. Thus, the messages sent and received by the platform consist of text-based documents written in XML (Extensible Markup Language) containing information about the building, its parameters and the actions to be performed.

The main disadvantage of this solution is the lower efficiency compared to other highly coupled implementations. Nevertheless, we consider that the possibility of an easy integration of different services based on different technologies compensates for this drawback.

A Building Monitoring Service has been implemented. It is described below to illustrate the Service Manager functionality. All the operations will be executed using SPARQL queries wrapped in XML messages. The main purpose of this Service is to request data from the different sensors present in the Energy Efficiency Research Facility (EERF) at CeDInt-UPM building and show them to the user graphically. These data can represent either real time information such as instant energy consumption or historical information, such as the annual average temperature in a given room.

Once the Service has been registered into the Bat-MP, it can perform a set of available actions over Logical Devices within the building. In case a Service wants to modify a certain Parameter which can be affected or controlled by more than one Physical Device (for instance, the parameter “Lighting” of a room could be modified either by switching up more lamps or by opening the electronic blinds, in case it is daylight), the Service will receive a message containing the type of all the Physical Devices which can be controlled to change this particular Parameter, (among a predefined set of Device Types the Service can receive when it is registered in the Bat-MP). Future versions of the Bat-MP will allow the Service to decide whether a certain Physical Device is used automatically or not.

When the Service first connects to the Bat-MP, it must be registered in the system through an SSL/TLS (Secure Sockets Layer/Transport Layer Security) connection. The complete authentication process is still under development by the time this paper is written. Providing that the Service has no information about the building in the first place, a SPARQL query will be sent to the Bat-MP asking about the rooms within it:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-
rdf-syntax-ns#>
PREFIX ro: <http://www.cedint.upm.es#>

SELECT ?r
WHERE{
  ?b rdf:type ro:Building.
  ?r rdf:type ro:Room.
  ?b ro:name "CeDInt-UPM".
  ?r ro:locateAt ?b
}
```

The Service will then receive a message containing the list of the rooms and their associated Uniform Resource Identifiers (URIs). If the service wants to know the available parameters inside a specific room with name “/CeDInt-UPM/rooms/show_room_1”, it will send the following query to the Bat-MP:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-
rdf-syntax-ns#>
PREFIX ro: <http://www.cedint.upm.es#>

SELECT ?d
WHERE{
  ?r rdf:type ro:Room.
  ?d rdf:type ro:Parameter.
  ?d ro:locateAt ?r.
  ?r ro:name “/CeDInt-
UPM/rooms/show_room_1”.
}
```

Once the Service receives the list of Parameters with their corresponding URIs, it can ask for the particular recorded values within a given period of time, or just for a real time value every time it changes. These values will be returned in the form of pairs “Date:Value” and codified using JSON (JavaScript Object Notation), a text-based standard format for data exchange through the Internet. In this case, since the developed power meters (BatMeters) are installed in the EERF, a message will be returned containing the Parameters representing the consumption of each electrical line, as well as other Parameters available in the room.

Finally, in order to get the consumption value of the HVAC system (with ID "HVAC") in a certain time (specified in milliseconds since January 1st 1970), the following message will be sent:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-
rdf-syntax-ns#>
PREFIX ro: <http://www.cedint.upm.es#>

SELECT ?v
WHERE{
  ?p rdf:type ro:Parameter.
  ?m rdf:type ro:Measure.
  ?p ro:id "HVAC".
  ?p ro:hasMeasure ?m.
  ?m ro:time "1338544800^^xsd:dateTime".
  ?m ro:value ?v
}
```

This simple example shows how it is possible to interact with different Physical Devices within a building considering only the physical quantities they are related to, and independently of the communication protocol they use.

Among the information which can be obtained by a service, the set of available actions can also be found. These actions correspond to the ones that can be performed considering the systems and devices present in a certain room. For instance, if a room has just a temperature sensor, the only available action related to the Parameter "Temperature" will be a method to get the current temperature. Whereas if there were also a HVAC system, the available actions would be a set of methods to get, increase or decrease temperature. In this context "available action" stands for an action that can be done considering the access permission of the user (not every user is allowed to perform certain kind of actions).

Thus, using a common description language like RDF combined with the implemented platform, a whole set of actions can be executed inside a building without needing to know highly specific details about the control systems present in the building.

To sum up, the Service Manager mainly provides an application programming interface allowing a Service to connect to the platform, letting the former register in the system and interact with the parameters in the building model. Besides, this layer supports access control security policies and resolves conflicts in concurrent accesses.

In order to provide scalability and flexibility, the system runs on OSGi framework, which allows the inclusion of new services in real-time and also new technology drivers and devices.

IV. BATNET (WIRELESS TRANSDUCER NETWORK)

In order to provide the building management platform with relevant data (such as power measurement) and to enable different applications control, a wireless transducer network (BatNet) has been designed and implemented. BatNet devices have been designed in order to collect building relevant data, send them to the Bat-MP, receive control and configuration orders from Bat-MP and finally perform the required actions within the building.

The main characteristics of BatNet, focused on avoiding the traditional systems limitations in terms of cost, interoperability, power consumption and collaboration, are:

a) 6LoWPAN-based network: 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) working group from the IETF describes the adaptation of IPv6 packets to IEEE 802.15.4 based networks. 6LoWPAN uses physical and data link layers based on IEEE 802.15.4 protocol (like other low power wireless communications as ZigBee or MiWi) and uses IPv6 (where a single IP address is assigned to each BatNet device) for the Network Layer [10].

b) CoAP communication interfaces: CoAP (Constrained Application Protocol) (currently an IETF draft) [11] is a specific web transfer protocol designed to be used with constrained nodes and networks. This protocol is simple and easy to implement.

c) Contiki Operating System: the open source operating system from SICS enables multitasking and implements both IPv6 and IEEE 802.15.4 standards under low-capacitance hardware requirements [12].

BatNet is based on a modular functional architecture, which eases the development of the different transducers and the integration of external devices. Each node of the BatNet comprises a processing and communications module (BatMote) plus a transducer module. The transducer module can be a power meter (BatMeter), ambient sensors (BatSense), power switch (BatPlug/BatSwitch), light dimmer (BatDimm) and blinds control (BatBlind) (see Fig.3).

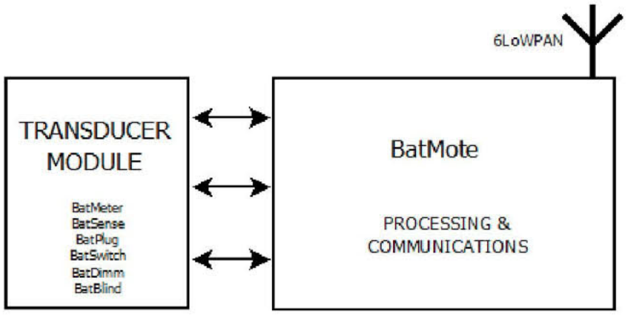

Figure 3: Modular architecture of BatNet devices.

A. PROCESSING AND COMMUNICATIONS MODULE

There are several processing and communication tasks common to every BatNet device independently of the node type (transducer, coordinator or repeater). Isolating these functions in a single module doesn't restrict the processing capabilities.

A low-cost and low-power processing and communication module has been designed and implemented: the BatMote (see Fig. 4).

The core of the BatMote is the ATmega 128RFA1 from Atmel Corporation. The ATmega 128RFA1 is a single chip microprocessor integrating an IEEE 802.15.4 transceiver, which allows a noticeable reduction of the hardware size. Besides, the ATmega 128RFA1 provides 8 ADC channels (10 bits resolution), allowing data collection from different sources simultaneously.

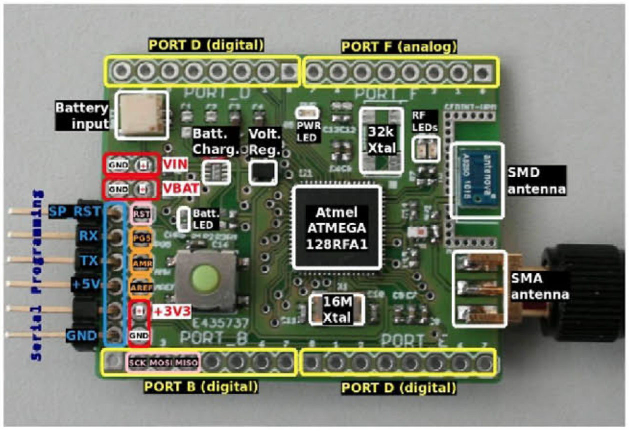

Figure 4: BatMote.

BatMote has been designed to allow easy connection of transducer modules, enable different installation locations and provide multiple data transmission intervals and sleep mode functions.

BatMote can be powered by batteries or other external power sources (3.3-12 V). It also includes a battery charger allowing the batteries to recharge through the use of a USB port.

BatNet devices can be located at electrical distribution board, room ceiling or walls; hence the size and shape of the antenna could be critic. BatMote is designed to be used with both SMD (Surface-mount technology) and SMA (SubMiniature version A) antennas.

B. TRANSDUCER MODULES

Depending on the functionality and signal type, different sensor and actuator modules have been designed:

a) BatSense: It is an ambient Multi-Sensor module which includes temperature, humidity, illumination and presence sensors together in a single device. Other ambient parameters (such as CO2 or noise) sensors can be added since extra pads and pins are available.

b) BatPlug and BatSwitch: BatPlug controls the ON/OFF status of appliances or devices connected to a socket while measuring the power consumption. A TRIAC (Triode for Alternating Current) deals with the ON/OFF switching. Power measurement is realized by monitoring both current (through a Hall Effect sensor) and voltage. BatSwitch can measure and control up to five appliances simultaneously.

c) BatMeter (power meter module):

It is a power meter sensor designed to be installed in the distribution board as a rail DIN (Deutsches Institut für Normung) device. It calculates simultaneously the electrical power and energy consumption of a maximum of seven electric lines (within the same phase) from real time voltage and current measurements (see Fig. 5).

The electric line voltage passes through a voltage transformer with two outputs. One output is used to power the BatMote passing through a voltage divider. The second output is used to measure the voltage signal. Using split core current transformers (SCT), up to seven different lines can be monitored. The measured analog signals enter the processor through the ADCs.

BatMote processor has been programmed to calculate different electric power parameters. The libraries from the Open Energy Monitor project[3] have been used and modified to run under the Contiki OS, in such a way that the device is able to measure more than one line at once.

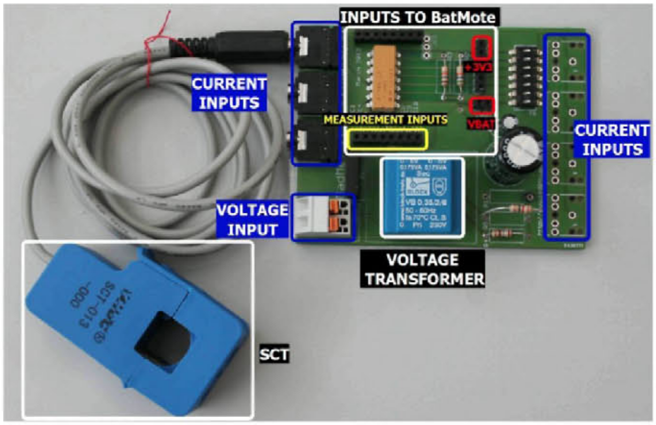

Figure 5: BatMeter and SCT.

The power and related parameters calculation works as follows:

1. BatMeter measures continuously instantaneous voltage and current, and BatMote calculates instantaneous power.
2. Every 20 electric signal periods (400ms) average voltage, current, active power, apparent power and power factor are calculated.

[3] Open Energy Monitor project: www.openenergymonitor.org.

3. Average values are sent every 15 seconds to a BatMote connected to a PC.

The whole smart meter consists of a BatMote and a BatMeter and has been installed in the distribution board at CeDInt-UPM Show-Room facility (see Fig. 6). This distribution board has separated lines to independently measure electrical consumption in three illumination groups, two HVAC systems and six appliance lines.

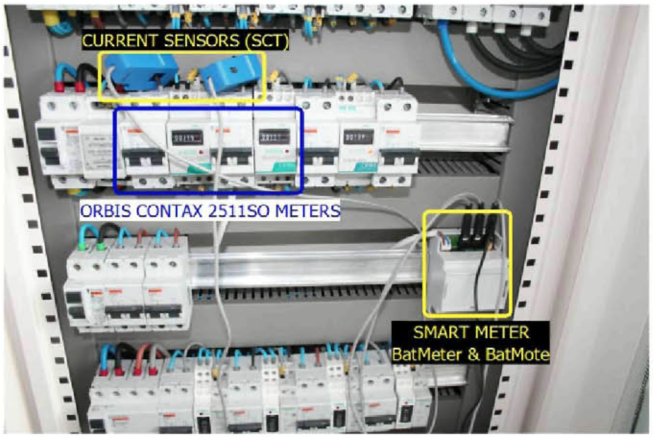

Figure 6: Smart meter installed at the distribution board

Different lines (Illumination, HVAC and Electric Water Heater) and appliances (Electric Heater and Computer) have been monitored.

To validate the system, energy consumption measures have been compared to those obtained from a commercial static meter installed at the distribution board (Model: Orbis Contax 2511SO). Table I shows the energy consumption measurements from different appliances during one hour working time:

TABLE I. ENERGY MEASUREMENTS RESULTS

Appliance	**Meter**	
	Orbis Contax 2511SO (Wh)	***BatMeter (Wh)***
Lights	310	314.6
Electric Water Heater	698	690.4
Electric Heater	1810	1796.3
Computer	63	60.45

As it can be seen in Table I, the developed smart meter shows accurate measurements for all the appliances.

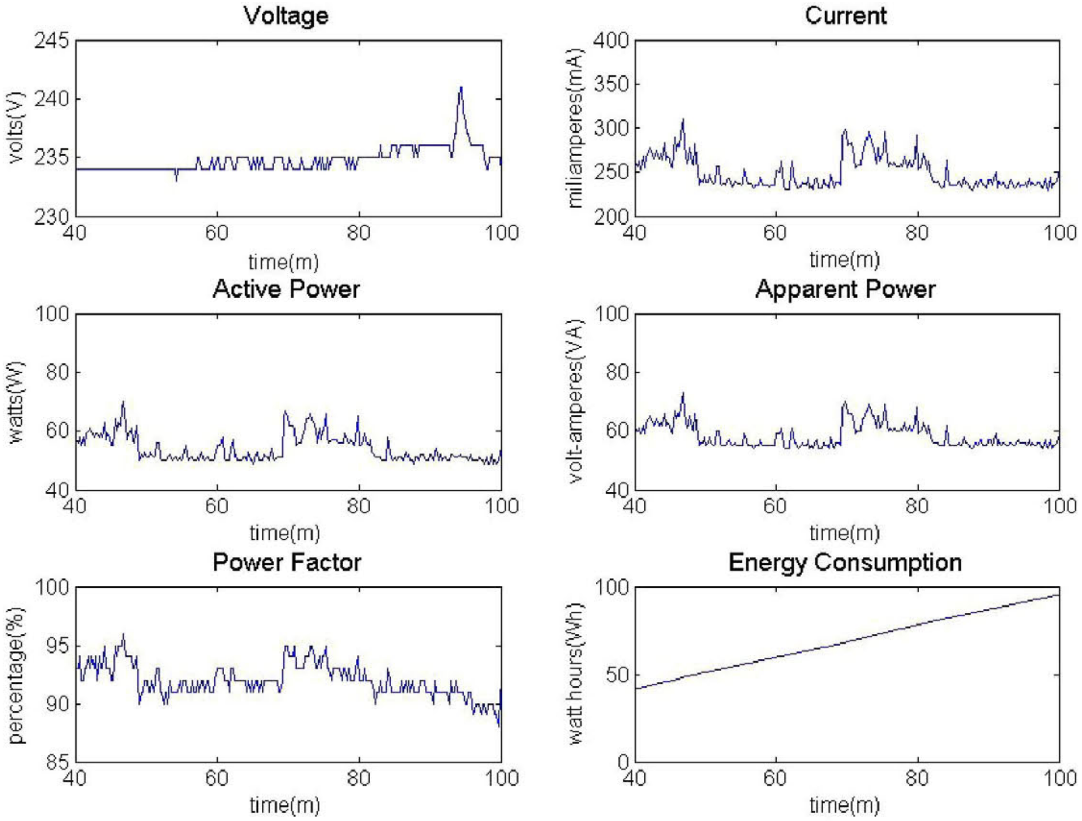

Figure 7: Voltage and Current measurements and Power (active and apparent), Power Factor and energy calculations for a PC.

Figure 7 shows the measured electric parameters (average voltage and average current) for a line where a PC has been connected. During the time period represented (1 hour), electric voltage varies between 233V and 237V. For long-time monitoring (a 96 hour period) a 228V-243V range has been recorded. Measured current is almost constant below 300mA for the PC dealing with low load processes during the time period.

Consequenty in the current measurements obtained, the active and apparent power do not suffer major oscilations. The power factor oscilates between 90% and 99%, which matchs the assumed power factor of a PC with an active power factor corrector power supply (95%). Energy consumption of 60 Wh corresponds to value measures for the expected value for 1.7 GHz Pentium IV computer.

V. CONCLUSIONS

In this paper an open multi-technology building energy management system is presented, which is useful to reduce building energy consumption. This system comprises a multi-technology management platform and a wireless transducer network and it is focused on solving current commercial systems problems: high cost, use difficulties and impossibility to add new services.

The management platform has been designed using an ontology-based approach in order to store information about the building structure and the devices within it. Hence, it is possible to design and implement different services which make use of these devices with no knowledge of technological details about the systems present in the building. A Java and OSGi based application has been implemented in order to integrate a building model and the different control systems and devices available inside it, providing also a framework to build all kind of services in an easy and technologically independent way.

A low cost and low energy consumption wireless transducer network based on 6LoWPAN has been implemented. The network consists of ambient sensors, power meters and actuators. The modular architecture design of the network nodes allows the easy integration of new sensors and actuators. BatNet Devices have been designed in order to be easily installed, in that they have a limited size and constant power supply is not required. Their modular architecture allows BatNet devices to measure several kinds of physical quantities within a building, being able to perform simple control actions as well.

A monitoring service to collect and show data from the building automation systems has also been implemented and tested in a real environment (at CeDInt Energy Efficiency Research Facility).

Finally, results of real time power consumption measurements realized by the power meter module are presented.

REFERENCES

[1] Luis Pérez-Lombard, José Ortiz, Christine Pout. “A Review on buildings energy consumption information”, Energy and Building, Elsevier, March 2007.

[2] Jeff Winston, "The Problem with Home Automation", Electronic House, 15 January 2008.

[3] Joern Ploennigs, Uwe Ryssel and Klaus Kabitzsch, “Performance Analysis of the EnOcean Wireless Sensor Network Protocol”, Proc. 20120 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pp 1-9, doi: 10.1109/ETFA.2010.5641313,2010.

[4] J.K.W. Wong, H. Lia and S.W. Wang, "Intelligent building research: a review," Automation in Construction, vol. 14, Issue 1, pp. 143-159, January 2005.

[5] A. Gómez-Pérez, M. Fernandez-López and O. Corcho, “Ontological Engineering with examples from the areas of Knowledge Management,” e-Commerce and the Semantic Web, Springer, 1st edition, 2004.

[6] Boris Malinowsky, Georg Neugschwandtner, Wolfgan Kastner. “Calimero: Next Generation” Automation Systems Group. Institute of Automation, Vienna University of Technology.

[7] Steffen Staab, Rudi Studer. “Handbook on Ontologies”. Springer, 2009.

[8] Georg Graham Klyne, Jeremy J. Carroll Resource Description Framework (RDF): Concepts and abstract Syntax. 2004.

[9] Prud'hommeaux, Eric; Seaborne, Andy). "SPARQL Query Language for RDF". *W3C*. World Wide Web Consortium.

[10] Lu Chia-Wen, Li Shu-Cheng and Q. Wu, “Interconnecting_ZigBee and_6LoWPAN_wireless_sensor_networks_for_smart_grid_applications,” Proc. 2011 Fifth International Conference on Sensing Technology (ICST), 2011, pp. 267-272, doi:10.1109/ICSensT.2011.6136979.

[11] Constrained Application Protocol (CoAP) draft https://datatracker.ietf.org/doc/draft-ietf-core-coap/

[12] George Oikonomou and Iain Phillips, “Experiences from porting the Contiki operating system to a popular hardware platform,” 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), pp 1-6, doi: 10.1109/DCOSS.2011.5982222.

[13] SPARQL Protocol for RDF http://www.w3.org/TR/rdf-sparql-protocol/. W3C Recommendation 15 January 2008.