Journal of Universal Computer Science, vol. 19, no. 17 (2013), 2543-2558
submitted: 15/2/13, accepted: 22/4/13, appeared: 1/11/13

Implementation of a Building Automation System Based on Semantic Modeling

Jaime Caffarel
(Universidad Politécnica de Madrid – CeDInt, Madrid, Spain
jcaffarel@cedint.upm.es)

Song Jie
(Universidad Politécnica de Madrid – CeDInt, Madrid, Spain
song.jie@cedint.upm.es)

Jorge Olloqui
(Universidad Politécnica de Madrid – CeDInt, Madrid, Spain
jolloqui@cedint.upm.es)

Rocío Martínez
(Universidad Politécnica de Madrid – CeDInt, Madrid, Spain
rmartinez@cedint.upm.es)

Asunción Santamaría
(Universidad Politécnica de Madrid – CeDInt, Madrid, Spain
asun@cedint.upm.es)

Abstract: This paper presents an Ontology-Based multi-technology platform designed to avoid some issues of Building Automation Systems. The platform allows the integration of several building automation protocols, eases the development and implementation of different kinds of services and allows sharing information related to the infrastructure and facilities within a building. The system has been implemented and tested in the Energy Efficiency Research Facility at CeDInt-UPM.

Keywords: Building Automation, Ontology, OSGi, RDF
Categories: H.1.2, M.4

1 Introduction

Ambient Intelligence (AmI) refers to the creation of environments embedded with sensing systems and pervasive devices in order to be responsive to the presence of people [José et al. 10]. In order to achieve this goal it is necessary to provide buildings with infrastructures and services which allow the interaction with all kinds of devices present within a building. Building Automation (BA) refers to the set of technologies and functionalities that allow monitoring and controlling all kinds of facilities within a building. These functionalities are related to AmI since Building Automation Systems (BAS) can serve as a basis for managing in a unified way different electrical equipments and more importantly, can act as a middleware for the development of services since they represent the lowest link point of all the various technologies usually present in automated buildings.

Among other characteristics, BAS are able to monitor data collected from sensors and control electrical devices and systems such as lighting or Heating Ventilation and Air Conditioning (HVAC). Nevertheless, available commercial systems present several issues. On the one hand, interconnection among devices using 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 [Pérez-Lombard et al. 07]. On the other hand, no additional external services can be added to these systems, limiting the development of future functionalities [Lihong 11]. Besides, commercial systems usually do not offer Application Programming Interfaces (APIs) in order to develop external applications or services.

In order to solve these issues, we propose an energy management system for buildings (Bat-MP), designed to ease the implementation of new applications and the integration of existing control systems within the building. The system mainly allows the development of different software services which can make use of the facilities present in the building. Bat-MP also enables the integration of different building control technologies, independently of the automation protocol used

The 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. Section 2 presents a general description of the building management system. In section 3, Bat-MP system architecture and its three main layers are detailed. In section 4, two implemented applications are described in order to test the feasibility of the platform with regards to energy applications. Finally, conclusions are presented in Section 5.

2 Related work

Regarding Home Automation, there are other existing systems such as [Reinisch et al. 10] which is a multi-agent system focused primarily on energy efficiency and user comfort, and [Xu et al. 09], which focuses on cooperative service composition within home environments. A similar approach is [Haya et al. 04], which also uses a blackboard structure (see below) and a unified model in order to store the data but it uses a distributed system approach instead of a centralized one. The system described in [Bonino et al. 08] is implemented using OSGi and makes use of an ontology to represent the building model and devices. Nevertheless, the actions available through the API are executed over physical devices, instead of over physical magnitudes (parameters) within the building, hindering technological abstraction.

Regarding the ontology used in different systems, [Ploennings et al. 12] describes a BAS ontology which abstracts the devices' technological details from their common functionalities. In [Sommaruga et al. 05] a very similar approach to the concept of parameter used in Bat-MP is described, since devices are seen as black boxes and only the necessary functions are exposed. In [Chen et al. 04], the authors integrate different existing ontologies, creating a comprehensive one. However, the application does not cope with the integration of different home automation protocols and technologies. In [Joo et al. 07], the authors create a basic ontology to describe the devices, environment and services in a Home Automation System. However, it does not consider the case that one device can be a sensor and an actuator at the same time and they can be used together or separately in different cases (for instance, some commercial thermostats combine sensing and control functionalities).

3 System description

The main objective of Bat-MP is to provide an open and easy platform to develop applications and services for managing the smart home or building. For this purpose, the system has been designed over a central concept: parameter. In Bat-MP a parameter is a data element that represents a state or value that a device is able to get or set in the real world, e.g. on/off switch, setpoint temperature, measured temperature or brightness level. These basic data elements provide a common object to implement the different functionalities provided from several Building Automation technologies, and simplifies the development of applications and services. Parameters' use creates an abstraction layer for developers by isolating devices functionalities from managing its underlying technologies.

Regarding home and Building Automation applications, a centralized control topology offers some advantages over a distributed one. For instance, a centralized control approach increases the security of the system and allows handling concurrency and possible conflicts between different users, devices or applications that try to access the same resources. Moreover, a centralized control element provides a way to store additional information about devices location and functionalities, the building and its occupants, which enables the elaboration of more complex and intelligent applications and services. Finally, a central control element is required to seamless integrate other devices that are based on different Building Automation technologies such as BACnet, KNX, LonWorks or ZigBee. Bat-MP is a middleware which can be used as a platform for the development of external applications in order to make use of the building facilities. On the one hand the system is able to communicate with devices from various Building Automation Technologies (connected to Bat-MP through an IP network) by means of different specific software modules. On the other hand, the system offers an abstraction layer of the devices and physical magnitudes within the building in order for external applications to interact with them. Figure 1 shows the System general structure.

Supported building automation protocols in Bat-MP are: LonWorks, KNX, ModBus, BACnet EnOcean, IPv6 over Low Power Personal Area Networks [1] (6LoWPAN) and X10, although it could also be expanded to support other ones.

[1] http://datatracker.ietf.org/wg/6lowpan/charter/

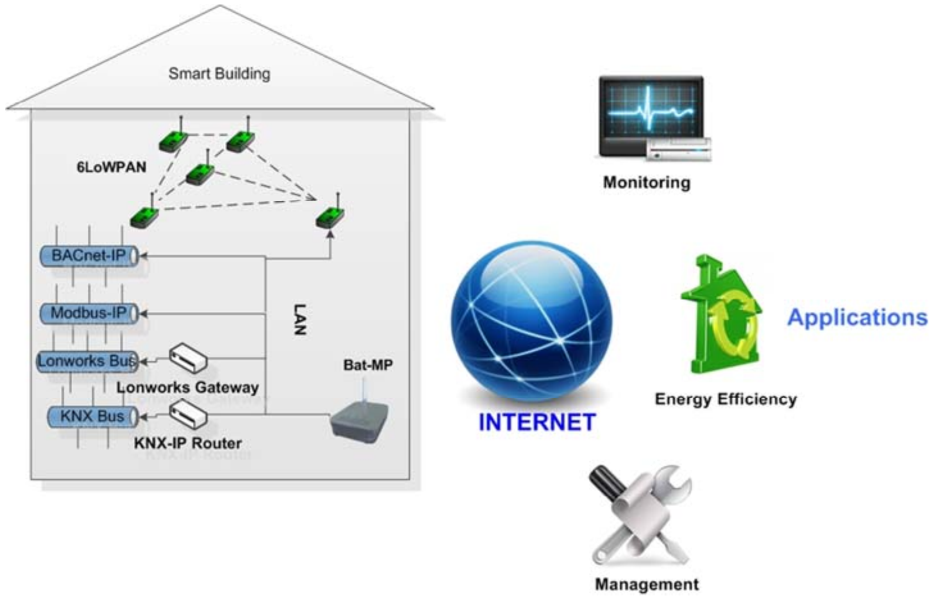

Figure 1: System architecture

4 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 an IP-based local area network. The system has been designed as a middleware aimed to support different kinds of applications 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 focused on energy efficiency within every building. This approach requires the middleware to solve the following issues:

- Allow interaction with different network protocols used in building automation, such as KNX, LonWorks, BACNet, or ModBus, in a unified way [Wong et al. 05].
- 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 [Gómez-Pérez 04].
- Allow third-parties to develop new applications which make use of the system in order to manage the building devices, needless to know low-level technological aspects.

To meet these requirements, the system has been designed in four layers and is implemented using a blackboard-based structure [Mullineux 91].

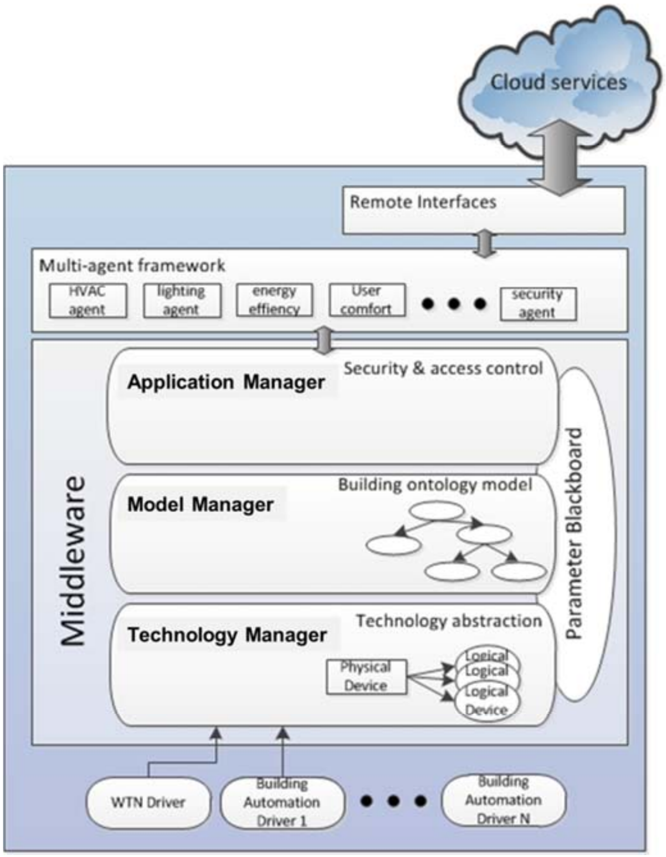

Figure 2: Platform software architecture

4.1 Parameter blackboard

This layer provides a common space for sharing parameter objects through the rest of the layers. Each parameter object triggers an event every time its value changes, providing an asynchronous communication between applications and technologies. When the Technology Manager receives a message from a Building Automation protocol informing that the state of a parameter has changed, it sets this new state in the parameter's value. This change causes an event that is captured by the Application Manager which informs the applications, and in the same way, whenever an application wants to set a parameter's value, the Technology Manager receives the event and sends a message to the corresponding Building Automation technology.

4.2 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 where the building is located). 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 Parameter represents a physical quantity or state (such as temperature, brightness level or on/off state) which can be directly modified by these devices. Technology Manager provides a way for external applications to interact with Parameters (i.e. physical magnitudes within the building) instead of over Physical or Logical Devices, abstracting the specific function details of each device.

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 [Erb et al. 07] 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. Besides, the Bat-MP allows interoperability with a 6LoWPAN [Kushalnagar et al. 07] wireless transducer network using Constrained Application Protocol (CoAP) as an application layer. 6LoWPAN is an adaptation layer to transmit header compressed IPv6 packets from over IEEE 802.15.4 networks, allowing IPv6 interoperability in low power and constrained devices. The use of this technology is becoming increasingly widespread among sensor and actuator networks [Han et al. 12]. On the other hand, Constrained Application Protocol [2] (CoAP) [Shelby et al. 11] provides a REST-based end-to-end communication protocol useful for constrained networks communication. CoAP is a lightweight protocol encapsulated in UDP which can be easily mapped to HTTP and implemented in Building Automation systems.

4.3 Model Manager

The amount of information that an intelligent building should manage includes the following:

---

[2] http://tools.ietf.org/html/draft-ietf-core-coap

- Building layout and 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, illumination level, electrical consumption or levels of gas, such as carbon monoxide.
- Information related to the use people make of the building, such as user timetable, specific usage of every area or user profile.
- Usage of the physical quantities by each application so that the system can make strategies about the priorities or the collaboration among them.

The use of ontologies to represent knowledge is becoming increasingly widespread [Staab et al. 09]. 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. Besides, the use of ontologies enables interoperability between specific knowledge domains and offers a way to apply reasoning and inference mechanisms to the data model [Hervás et al. 10]. Furthermore, the use of ontologies allows the further publication, sharing and extension of the data model.

The Model Manager uses a Resource Description Framework (RDF) data model, which is a standard model for data interchange on the Internet which uses statements in the form of subject-predicate-object expressions, to describe all the entities, characteristics and properties of a building. An ontology-based model of the building has been developed in order to store all the information described above.

Existing ontologies such as Semantic Sensor Network [3] (SSN) are very specific and focused on sensor devices. Building Information Models (BIM) are vendor-neutral exchange standard formats which are used to represent a highly precise model of a building, containing structural and functional characteristics of it. Nevertheless, their applicability regarding ontology-based systems present some issues such as the difficult implementation due to the amount of information needed to be taken into account [Guangbin et al. 11] or the lack of important concepts for energy analysis [Kofler et al. 10]. The ontology used to define the building contains all the necessary concepts describing the building layout and facilities. The main concepts of the model are:

- *Space*. A space within a building is an enclosed area by means of walls, windows, ceilings and floors. Every space has its own physical characteristics, such as size and geographical orientation (in case the space has windows, the direction they are facing). In order to simplify the modeling process of the building, the location of every space within it is only defined by the adjacent space and their orientation.

---

[3] http://www.w3.org/2005/Incubator/ssn/wiki/Semantic_Sensor_Net_Ontology

- *Zone*. Sometimes, the spaces within a building are too extensive and a finer granularity is required. A zone represents a part of a space, which is managed independently from the other ones.

- *Device*. One device can be either a sensor (which gives a measurement of a physical quantity), an actuator (which can perform actions in order to change certain magnitudes inside a room, such as a light dimmer), or both, combining different functionalities. One room can contain zero or more devices. This concept of Device is corresponding to the "Logical Device" provided by the Technology manager.

- *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 or off but not dimmed, only those two actions will be available in the parameter.

- *Measurement.* Each measurement represents a pair (value, timestamp) which stores the physical value of the parameter in a given timestamp.

This ontology describes devices and observations and all its associated concepts, focusing on the physical magnitudes and the actions that can be performed within each space. 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 Recommendation [4] when implementing ontology databases.

4.4 Application Manager

The main purpose of this layer is to provide the different external services (Cloud 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 application is to control the temperature of a certain room or area, the application 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 application, the Application Manager also provides a common interface to access to the Management Platform through the Internet. Thus, a web application has been implemented using a Service-

---

[4] http://www.w3.org/TR/rdf-sparql-query/

Oriented Architecture (SOA). In this context, an application is every platform-independent program which interacts remotely with parameters and devices within the building through the interface provided by the Bat-MP. The SOA model is based on a message exchange process between the entities involved and allows a loosely coupled interaction between the applications and the platform. The main goal of this design model is to achieve an easier integration of different applications, using defined protocols to communicate with them. This approach gives the developer liberty to design applications using different technologies or programming languages, since the only communication requirement is the format of the messages to be exchanged. Particularly, Bat-MP messages consist of text-based documents written in XML containing information about the building, its parameters and the actions to be performed.

The main disadvantage of this approach is the lower efficiency compared to other highly coupled implementations, among other reasons, because a SOA architecture requires the use of longer messages between communication endpoints (in the case of Bat-MP, due to the XML format of the messages), thereby introducing a transmission overhead. Nevertheless, we consider that the possibility of an easy integration of different applications based on different technologies compensates for this drawback.

Once an application registers itself within the system, it can read or modify certain parameters according to the permissions the system administrator gives to it. In order to provide high flexibility to the application design, these permissions are explicitly assigned to three different objects: parameters, spaces within the building (each application has read or write permissions over particular building areas) and user role (not every user of the application can perform the same operations over parameters or spaces). The actions that an application can perform over a certain space or zone are also conditioned to the equipment available in that area. For instance, if a space has just a temperature sensor, the only available action related to the Parameter "temperature" will be a method to get the current temperature. If there were also a HVAC system, the available actions would be a set of methods to get, increase or decrease temperature.

The public interface offered by the Application Manager can be accessed using two different message formats, which are used mainly to describe actions over the parameters within the system.

The first method uses SPARQL queries to retrieve information about the parameters. When using this method, the application gets the ontology model of the building written in RDF so that it can perform queries over it. These RDF queries can be sent to the Application Manager in order to obtain information related to the building model and the parameters available. 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.

The second method uses a format based on the Green Building XML[5] schema (gbXML). This schema is a non-proprietary XML-based standard designed to describe all the information about the building as well as the systems and devices

[5] http://www.gbxml.org/

inside it. The standard uses XML objects to describe properties of the building (geographical location, number of floors or surface area), information about physical magnitudes (temperature, lighting, etc.) and information of all the systems within the building. The Application Manager implements a REpresentational State Transfer (REST) interface in order for the external application to read information about parameters or spaces and to send control orders to interact with these parameters. By means of the definition of objects codified using this schema combined with the Application Manager interface it is possible for an external application to interact with the parameters it wants to read or modify.

The format used by Bat-MP to send measurements from the building devices and systems to the external applications is Sensor Markup Language [6] (SenML), which is a data model designed to represent simple measurements and information about devices. SenML was designed in order for constrained devices to send simple encoded sensor measurements, and it can be easily mapped to JavaScript Object Notation (JSON), XML and Efficient XML Interchange (EXI). In SenML, each message is structured as a single object with attributes containing an array of entries. Each entry contains the parameter identifier, the current value and the time when the measurement was taken. For instance, the following message represents the value of the parameters “current” and “voltage” at an unspecified time.

```
{"e":[
     { "n": "voltage", "u": "V", "v": 119.8 },
     { "n": "current", "u": "A", "v": 0.9 }],
}
```

Although SenML is not suitable for representing highly complex data, this is not an issue due to the fact that more specific metadata about the the parameters can be described using SPARQL or gbXML prior to the data exchange, thus simplifying and streamlining the data flow between Bat-MP and the external application. Finally, thanks to their simplicity, SenML messages can be easily parsed using available libraries from several object oriented programming languages.

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

The system runs on the Equinox platform, an Open System Gateway initiative (OSGi) implementation. This platform is based on a Java environment, allowing to run the software over different operating systems and machines. The main benefit of using OSGi specification is to create modular and scalable software elements called Bundles. Bundles can be added or removed in run-time, where OSGi tools manage its life-cycle and the security issues between them [Tao et al. 04]. This approach is used in BatMP to add or remove dynamically new devices, technologies or applications.

[6] http://tools.ietf.org/html/draft-jennings-senml-10

5 Applications

In order to test the feasibility of the platform, two independent applications have been deployed.

5.1 Monitoring Service

The first application offers a simple monitoring service of the parameters and devices within the building. This web-based application is described to illustrate a basic Application Manager functionality. All the operations will be executed using SPARQL queries which are sent to the Bat-MP. The main purpose of this Application is to request data of 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.

Providing that the Application has no information about the building in the first place, a SPARQL query will be sent to the platform 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/residentialontology.owl#>
SELECT ?r
WHERE{
  ?b rdf:type ro:Building.
  ?r rdf:type ro:Room.
  ?b ro:name "CeDInt-
  UPM"^^<http://www.w3.org/2001/XMLSchema#string>.
  ?r ro:locateAt ?b
}
```

The Application will then receive a message containing the list of the rooms and their associated Uniform Resource Identifiers (URIs). If the application 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:

```
SELECT ?p
WHERE{
  ?r rdf:type ro:Room.
  ?p rdf:type ro:Parameter.
  ?p ro:locateAt ?r.
  ?r ro:name "/CeDInt-
UPM/rooms/show_room_1"^^<http://www.w3.org/2001/XMLSchema#string
>
}
```

Once the Application 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, a text-based standard format for data exchange through the Internet.

Finally, in order to get the consumption value of the HVAC system (with ID “HVAC”) in a certain period of time, the following message will be sent:

```
SELECT ?v
 WHERE{
   ?p rdf:type ro:Parameter.
   ?p ro:id "HVAC".
   ?m rdf:type ro:Measure.
   ?p ro:hasMeasure ?m.
   ?m ro:time ?time.
   ?m ro:value ?v.
   FILTER (
      xsd:dateTime(?time)>="2012-06-13T00:00:00Z"^^xsd:dateTime
      && xsd:dateTime(?time)<="2012-06-
30T23:59:59Z"^^xsd:dateTime
   )}
 ORDER BY ?time
}
```

Using these messages, the application graphically shows the user the list of parameters available in the building as well as their values during a certain period of time. The application has been developed using Java Server Pages (JSP) and JavaScript in order to display the graphs. When the application is started and connected to Bat-MP, it shows all the available parameters within the building, structured by floors and rooms. The given parameters are associated to devices using different home automation technologies such as KNX, LonWorks and BACnet. However, the user is not concerned about these technological details.

Once the parameters are shown, the user can select the ones they want to monitor (dragging them into another area). Figure 3 represents this interface.

Once the parameters have been selected, their stored values are shown graphically within an appropriate timescale. Figure 4 shows this interface.

This 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.

5.2 Climate Control Application (ClimApp)

The second application offers a more complex functionality consisting in a climate control service. The climate control application ClimApp [Gómez-Otero et al. 12] manages HVAC systems in buildings in order to increase user comfort while reducing energy consumption. ClimApp has been installed at CeDInt building. Using information from various data sources (temperature and humidity sensors connected through KNX on every room and user temperature preferences), ClimApp calculates optimum temperature in every room of the building and sends control orders to the HVAC systems managed through BACnet protocol. ClimApp uses data from KNX temperature and humidity sensors within CeDInt facilities. To get these data from the sensors, a specific KNX protocol implementation is used. If a set of new sensors using another technology such as LonWorks were available to monitor room ambient parameters, its control technology would have to be explicitly integrated in ClimApp.

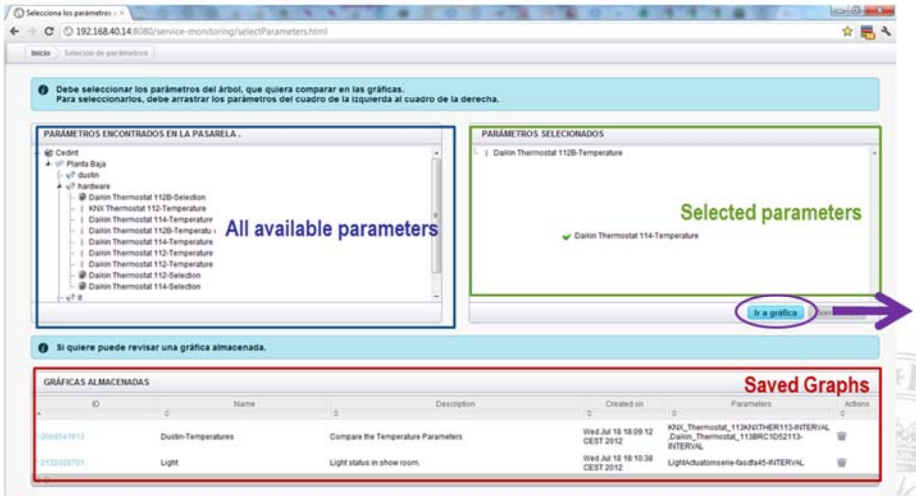

Figure 3: Monitoring application. Main interface

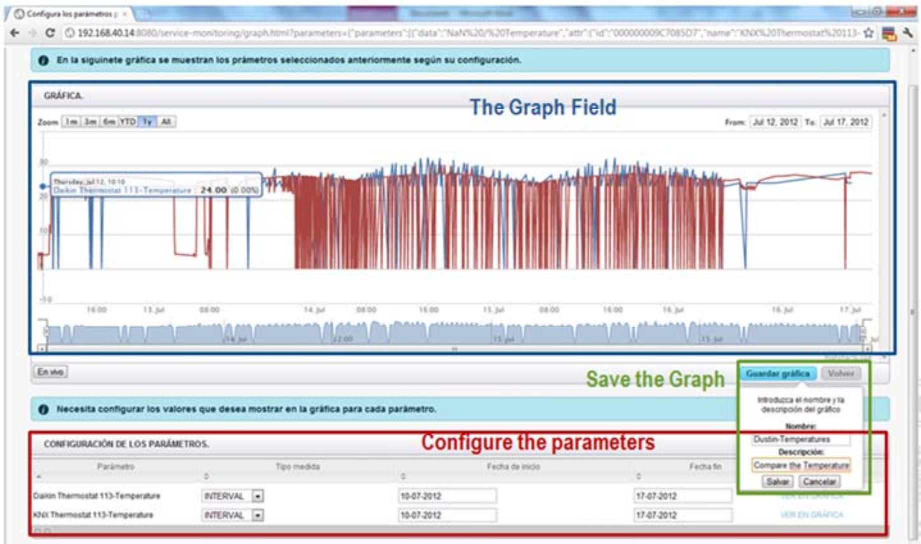

Figure 4: Monitoring application. Graph interface

In order to avoid further developments for the integration of new technologies such as LonWorks in the ClimApp and to test the feasibility of the communication between a complex external application such as ClimApp and Bat-MP, both systems have been integrated. ClimApp has been modified so that instead of checking specific KNX sensors, it registers itself within Bat-MP, uses its REST interface and retrieves these parameter values using the SenML format (temperature and humidity of the rooms) with no need to deal with the KNX protocol. Besides, ClimApp obtains temperature values from the outside of the building through a small wireless weather station connected using the IEEE 802.15.4 protocol. This parameter (exterior ambient temperature) is obtained transparently from the underlying technology. Figure 5 shows the integration of both systems.

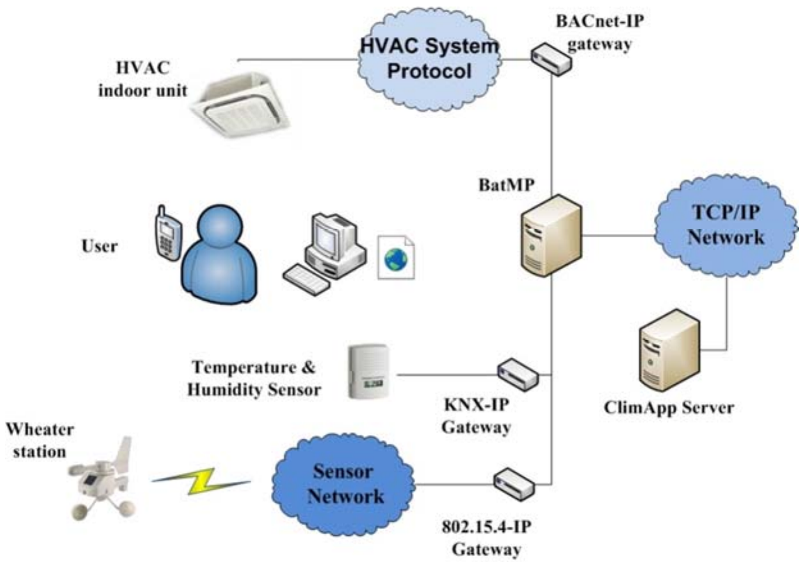

Figure 5: ClimApp integration in Bat-MP

These applications show the possibility of using an ontology-based building management system within a real environment, offering on the one hand a way to interact with heterogeneous devices transparently of the underlying technologies involved and on the other hand, a middleware to develop different applications using those technologies.

6 Conclusions

This paper presents an open multi-technology energy management platform useful to manage different building automation technologies. The platform tackles current commercial system issues: interconnection between different Building Automation technologies and the possibility of adding new services or applications in order to manage them.

The system has been designed using an ontology-based approach in order to store information about the building structure and the devices present within it. This ontology makes use of parameter concept. Hence, it is possible to design and implement different services making 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.

Two different applications have been implemented or adapted in order to test the Bat-MP within a real environment. On the one hand, a monitoring application to collect and show data from the building automation systems has been implemented and tested in a real environment: EERF at CeDInt-UPM. On the other hand, a climate control application using KNX and BACnet protocols has been adapted in order to take advantage of the Bat-MP capabilities.

References

[Bonino et al. 08] Dario Bonino, Emiliano Castellina, Fulvio Corno. "DOG: an Ontology-Powered OSGi Domotic Gateway". 20th IEEE International Conference on Tools with Artifitial Intelligence.

[Chen et al. 04] Chen, H.; Perich, F.; Finin, T.; Joshi, A., "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications". 2004. The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services.

[Erb et al. 07] Erb B, Malinowski B, Neugschwandtner G, Kastner W, Granzer W (2007) "Calimero Project". http://sourceforge.net/projects/calimero.

[Gómez-Otero et al. 12]C. Gómez-Otero, R. Martinez, J. Caffarel "ClimApp: A Novel Approach of an Intelligent HVAC Control System" 2012 WISIS.

[Gómez-Pérez 04] 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". 2004. Springer, 1st edition.

[Guangbin et al. 11] Wang Guangbin et al. "Research on some critical problems of contracting for Building Information Model". 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet).

[Han et al. 12] Donghyuk Han, Jong-Moon Chung, and Raymond C. Garcia. "Energy Efficient Wireless Sensor Networks based on 6LoWPAN and Virtual MIMO Technology". 2012 Sch. of Electr. & Electron. Eng., Yonsei Univ., Seoul, South Korea.

[Haya et al. 04] Haya, P.A., G. Montoro, X. Alamán, 2004. "A prototype of a context-based architecture for intelligent home environments". CoopIS04, *Lecture Notes in Computer Science (LNCS)*, Vol. 3290, pág. 477-491.

[Hervás et al. 10] Ramón Hervás, José Bravo, Jesús Fontecha. "A Context Model based on Ontological Languages: a Proposal for Information Visualization". J.UCS (Journal for Universal Computer Science), vol. 16, no.12.

[Jian et al. 08] He Jian, Dong Yumin, Zhang Yong, Huang Zhangqin. "Creating an Ambient-Intelligence Environment Using Multi-Agent System". 2008 International Conference on Embedded Software and Systems Symposia.

[Joo et al.07] Inbong Joo and Jongyoul Park and Euihyun Paik."Developing Ontology for Intelligent Home Service Framework". ISCE 2007. IEEE International Symposium on Consumer Electronics

[José et al. 10] Rui José, Helena Rodrigues, Nuno Otero. “Ambient Intelligence: Beyond the Inspiring Vision”. J.UCS (Journal for Universal Computer Science) vol. 16, no. 12 (2010). 1480-1499.

[Kofler et al. 10] Mario J. Kofler, Wolfgang Kastner. “A Knowledge Base for Energy-Efficient Smart Homes”. 2010 IEEE International Energy Conference.

[Kushalnagar et al. 07] N. Kushalnagar, G. Montenegro and C. Schumacher, 2007. “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement and Goals”. IETF RFC 4919.

[Lihong 11] Liu Lihong. “Study on Problems Existing in Building Intelligent Community and System Integrated Management in China”. 2011. International Conference on Business Management and Electronic Information (BMEI).

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

[Mullineux 91] Glen Mullineux. “A Blackboard Structure for Handling Engineering Design Data”. 1991. Engineering with Computers 7 (1991), 185-195. Springer.

[Pérez-Lombard et al. 07] Luis Pérez-Lombard, José Ortiz, Christine Pout. “A Review on buildings energy consumption information, Energy and Building”. Elsevier, March 2007.

[Ploennings et al. 12] Joern Ploennings, Burkhard Hensel, Henrik Dibowski, Klaus Kabitzsch. “BASont – A modular, adaptative Building Automation System Ontology”. 2012. 38th Annual Conference on IEEE Industrial Electronics Society. (IECON 2012).

[Reinisch et al. 10] Christian Reinisch, Mario J. Kofler, Wolfgang Kastner. “ThinkHome: A Smart Home as Digital Ecosystem”. 2010. 4th IEEE International Conference on Digital Ecosystems and Technologies.

[Shelby et al. 11] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, 2011. “Constrained Application Protocol (CoAP)”. Internet-Draft (work in progress) [Online].

[Staab et al. 09] Steffen Staab, Rudi Studer. Springer. “Handbook on Ontologies.” Springer. 2009.

[Sommaruga et al. 05] Lorenzo Sommaruga, Antonio Perri, Francesco Furfari. “DomoML-env: an ontology for Human Home Interaction”. 2005. 2nd Italian Semantic Web Workshop (SWAP 2005).

[Tao et al. 04] Gu, Tao, Hung Keng Pung, Da Qing Zhang. “Toward an OSGi-based infrastructure for context-aware applications” 2004. Pervasive Computing, IEEE.

[Wong et al. 05] J.K.W. Wong, H. Lia and S.W. Wang “Intelligent building research: a review”. 2005. Automation in Construction, vol. 14, Issue 1, pp. 143-159.