An IFC4-based Middleware for Data Interoperability in Energy Building Operation

José L. Hernández, Susana Martín and César Valmaseda
Fundación CARTIF, Parque Tecnológico de Boecillo, Valladolid, Spain

Keywords: Middleware, Interoperability, Data-model, IFC4, Heterogeneous Data, Interfaces, Data Consistency.

Abstract: This paper addresses the existing gap in data interoperability among heterogeneous resources for energy service systems of building automation. In this sense, the middleware is the core of the communication between heterogeneous data samples and the application services. This kind of solutions integrates the multiple building data resources to gather the information within the context of energy and buildings and couples the data in a single signal. The middleware also manages the data in an harmonized way by means of the representation of the information in a well-established data-model as IFC4 which is widely used in the building topic. This kind of harmonic communication allows the exchange of information among the entities in complex platforms by common formats in order to ease the interpretation of data. Then, interoperability is a key factor for achieving connectivity that is reached in the present middleware though the event-driven communication mechanisms and the well-known interfaces.

1 INTRODUCTION

Nowadays, Building Energy Management Systems (BEMS) are a growing field of interest taking into account it is estimated that up to 40% of total energy in Europe is consumed by buildings (Sustainable Buildings & Climate Initiative, 2009). With the objective of reducing the consumption, several research efforts and novel technologies are underway towards designing smart systems (C21, 2014)(NET, 2014). Within this context, the BaaS (Building as a Service) project aims at reducing 15% of energy consumption through ICT tools and by adapting control decisions in real-time (Baa, 2014). For such purpose, it is strengthened the intersection of control algorithms, thermal simulation, communication technologies, middleware platforms, and building technologies.

As core of these systems, middleware enables the communication between the building physics (i.e. data measurements) and the application services (assessment, prediction and control). The objective of using a middleware is to obtain interoperability, transparency and coherency in the communication of these heterogeneous entities. In this scenario, the BaaS middleware presented in this paper offers data management from various in- and out-of-building sources that act as central repositories for all static and dynamic building data (Baa, 2014). Other external ICT systems like a weather station are also considered as data sources.

On the other hand, within the building context, the aforementioned efforts still lack the full interoperability concept. To achieve it, data models for the representation of this information are required, as depicted in (Crosbie et al., 2011), where semantic integration of energy information is highlighted. Then, creating a common language helps the integration and interoperability among entities along the building life cycle. One of these data model is the Industry Foundation Classes (IFC - ISO 16739:2013) (IFC, 2014) which defines a set of domains where the building facilities are modeled within the architecture, engineering and construction (AEC) industry. BaaS system has considered this IFC4 specification for the representation of the information in all its layers. With this approach, building stakeholders are able to interpret data in the design of assessment and optimization tools.

Therefore, this paper presents a novel IFC4-based middleware for the integration and coupling of heterogeneous building data in a homogeneous way. The main purpose is to offer to the end user interoperability and transparency among all the building automation entities through the same vocabulary. With that aim, it abstracts the building physics in order to provide two-way communication between the data sources and the high level services. Besides that,

L. Hernández J., Martín S. and Valmaseda C..
An IFC4-based Middleware for Data Interoperability in Energy Building Operation.
DOI: 10.5220/0005371002870294
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 287-294
ISBN: 978-989-758-096-3

the middleware has been designed according to standard and open source projects, such as OSGi (Open Services Gateway initiative), to provide a framework to Service-Oriented Architectures (SOA) and event communication.

To overcome the description of the BaaS middleware, this paper is mainly organized in three sections where the requirements of the specific BaaS systems are pointed out and comparison with existing platforms is realized. Next, the high-level architecture, communication interfaces, data model and interoperability concepts are explained. Finally, the distributed deployment of the system is depicted before the final conclusions reached during this study.

2 BaaS MIDDLEWARE REQUIREMENTS

The first step in any software development must be the compilation of all the requirements and functionalities which should be assured. BaaS design is not an exception of this methodology and the functionalities and performance requirements have been collected. Table 1 summarizes these functional (FR) and non-functional (NFR) requirements.

First of all, any middleware, by definition, has to ensure interoperability among all the components which is easier reached by means of making use of standards or recommendations offered by the different standardization bodies. In the specific case of BaaS, the middleware needs to access to multiple services and heterogeneous pieces of information as described in the next bullets:

- Building Management Systems (BMS)
- DataWarehouse (DWH)
- Building Information Model (BIM)
- Existing ICT systems and external data services
- Assessment, Prediction and Optimal-Control services and analytics

Each piece of information has its own communication language, and it has to be merged into a common one. Therefore, the middleware should be able to translate and adapt the different information representation into a same vocabulary. For such purpose, the ISO 16739:2013 standard (IFC4) (IFC, 2014) is selected. Then, the middleware should communicate the specific protocol of the BMS and obtain sensor data translated into IFC4. This is also applicable to the external services (e.g. weather forecast or weather station) whose information is required from the high level services. Moreover, the DWH (Mo et al.,) and BIM Server (TNO, 2010) are IFC4-based designed. The first one is the main repository of the BaaS system for dynamic data and the BIM is in charge of managing the building static data.

Table 1: Requirements summary.

Identifier	Requirement description
FR-1	Baas MW should assure interoperability among all the entities involved in energy assessment
FR-2	BaaS MW should represent data into the same vocabulary or data model
FR-3	BaaS MW should implement the specific communication protocol at building level
FR-4	BaaS MW should connect Web Service for accessing Weather forecast
FR-5	BaaS MW should manage BIM entities for the representation of the building physics
FR-6	BaaS MW should maintain a coherent data repository with dynamic data
FR-7	BaaS MW should maintain coherency and consistency among all the data sources
NFR-1	BaaS MW should assure Quality of Service parameters
NFR-2	BaaS MW should be compliant with cloud deployments

In order to enable the communication with the aforementioned data-sources, a set of well-established communication interfaces are defined based on standards (Hernandez et al., 2013b). However, each resource heterogeneously accesses to the information, hence, the middleware covers this gap with these communication interfaces, which transparently translates the data into IFC4. This solution assures the interoperability and integration of heterogeneous data in an homogeneous way. Besides that, the information between multiple data sources must be coherent, therefore, additional components which ensure data consistency are required.

Last but not least, some performance requirements (non-functional) have been defined to assure Quality of Service. In that way, the system should provide a sufficiently high availability (service level agreement) and be scalable and replicable as well as fault-resilient. Besides that, the BaaS system is designed and developed according to cloud computing premises which allow deploy the Building as a Service or Platform as a Service.

2.1 Fitting the Requirements in Existing Middlewares

After detailing the requirements and specifications for BaaS middleware, the next phase is to determine if any existing middleware overcomes with these needs. Within the European context, multiple projects are carried out, such as Campus21 (C21, 2014) or MOST (Zach et al., 2012), but also other research initiative, such as the one presented in (MWs, 2013), define middleware platforms for buildings. Moreover, several commercial solutions are in the marketplace, as for instance NETxAutomation (NET, 2014) or Smarkia (Sma, 2014). All of them treat the building communication topic from an energy point of view and using middleware platforms with different features as explained below, although the commercial solutions go beyond the middleware platform providing the complete Building Energy Management System.

Starting with the Campus21 middleware (Schulke et al., 2013), its objective is to render building operations as happening in BaaS project. It also considers multiple data sources whose information is exchanged among multiple entities and/or components. Moreover, the requirements are similar to the BaaS one (Schulke et al., 2013). However, it does not apply with the common data model because it shares the data in Java-based objects. Even more, the protocol for accessing the BMS is BACnet/IP, whereas BaaS requires adaptability.

Next, Flotyski et al (MWs, 2013) present a building middleware working in the Internet of Things so as to integrate multiple data sources in an OSGi framework. This middleware is developed based on events and as a gateway for connecting heterogeneous devices in a modular manner, similar to BaaS. However, it is only centered into the integration of devices though modules (MWs, 2013) without allowing high-level services for the facility management. Even more, each module is in charge of the specific device, whereas BaaS middleware adapts the information into the data model before exchanging it (Hernandez et al., 2013b).

Regarding the Smarkia system (Sma, 2014), it is a complete Building Energy Management System which follows a similar architecture than BaaS because it gathers the information from the building level so that high-level services perform the analytics. Yet, it does not use any specific data model for exchanging the information, but the internal communication mechanisms interprets these data. The disadvantage of Smarkia, in comparison with BaaS, is that the scalability is reduced.

Finally, NETxAutomation has developed the Voyager 5.0 system (NET, 2014) which is a system which integrates multiple protocols to access the BMSs. It also applies control algorithms to improve the energy performance in buildings, as presented in the demonstrator cases (NET, 2014). Nevertheless, it is a closed solution without the possibility of scalability which does not couple multiple data sources, but it is based on BMS. As well, it does not implement any data model for the representation of the information.

2.2 Beyond the State of the Art

Once having explained how existing middlewares fit with the requirements of the BaaS system, it can be concluded that no one applies with the full stock of requirements in the BaaS system, as pointed in the Table 2. This table highlights the differences from the presented middlewares in comparison to the requirements and draws why BaaS platform goes a step forward in order to cover the lacks of existing systems.

Above of all, the most important result of the middleware is the presence of a common context for exchanging data. IFC4 has been selected as the data model for the BaaS middleware, and although this standard is large, a subset of objects have been used, as described in section 3.2. BaaS is able to map the information from the data sources in this common format. The great novelty of that is the capability to represent the information in the same context of the application, i.e. buildings.

On the other hand, BaaS manage multiples and heterogeneous data sources, such as Building Management System, BIM, DataWarehouse, etc., as depicted in section 3.1 and 3.3. Then, so as to increase the scalability and replicability of the solution, the presented solution connects the communication protocols of the building, instead of providing a communication interface and having to adapt the building to the software. For example, Campus21 middleware is compliant with BACnet, but the full stock of buildings are not BACnet compatible. With BaaS, the connectivity with open and standards protocols is assured. As well, the integration of additional data resources eases the common understanding of the energy performance in buildings through the building information and weather conditions, among others.

Another advantage of the BaaS middleware is the coherency of data among data sources, as explained in section 3.3. Normally, the middleware systems ensure the interoperability among different entities, but it does not take care of the coherency of data. This functionality is usually implemented in high level services or by means of intelligence in the data layer. Nevertheless, the novelty presented in BaaS, is the ca-

Table 2: Comparison among existing middlewares.

Req.	Campus21	Flotyski	Smakia	NETx
FR-1	Yes	Yes	Yes	Yes
FR-2	No	No	No	No
FR-3	No	Yes	Yes	Yes
FR-4	Yes	Yes	Yes	Yes
FR-5	Yes	No	No	No
FR-6	Yes	Yes	Yes	Yes
FR-7	No	No	No	No
NFR-1	Yes	Yes	Yes	Yes
NFR-2	No	No	Yes	Yes

pability of the middleware to periodically check the data sources, compare them and determine if any inconsistency has been detected so that the administrator could be automatically reported about it.

Last but not least, cloud consideration are not always taken into consideration in middlewares, but the capability of providing interoperability. Taking into account this current trend in novel systems, BaaS has been designed to operate in cloud platforms and offers the opportunity of being deployed under Building as a Service where the building is the core.

3 ARCHITECTURE OF MIDDLEWARE-BASED BaaS SYSTEM

The BaaS middleware has been designed according to the functional and non-functional requirements described in the section before. With all of them in mind, the high-level architecture of the BaaS middleware is an event-driven platform following the Service-Oriented Architecture (SOA) patterns and the OSGi framework. Figure 1 (Hernandez et al., 2013b)(Floeck et al., 2013) represents this system which is composed by three levels: data layer, communication logic layer (CLL) or middleware layer and the application layer based on high-level services (assessment, prediction and optimized control). Moreover, the communication interfaces are drawn which ensure the interoperability among all the components of the system, as well as they integrate the heterogeneous data sources.

The Communication Logic Layer (CLL) of this architecture, in the form of a middleware entity, is the topic of the present paper. It is basically split into two sub-layers: the core communication and data access object (DAO) (Hernandez et al., 2013b). Communication core is responsible for the management of the signals and the business core. In contrast, DAO aims the integration of heterogeneous data sources and homogeneizes the information in a common data model.

Apart from that, the CLL is, at the same time, divided into two entities: Data Acquisition and Control Manager (DACM) and Domain Controller (DC). The first one is the server side and manages the communication procedure and the second one is in charge of the building connectivity. The objective of this desing is to increase the scalability and extensibility, as well as allow the cloud deployment.

3.1 Interoperability Communication Interfaces

Interoperability is a concept inherent to middleware platforms and it describes the capability for interconnecting modules in a transparent way so as to exchange information. Communication interfaces are required here to set up a well-known communication mechanism focused on allowing connectivity, performing the expected functionality, and adaptating communication languages between layers.

In the context of the BaaS middleware, several interfaces (I1 to I10) have been defined (Hernandez et al., 2013b)(Floeck et al., 2013) (see Figure 1). I-1 is the interface between the application layer and the middleware itself to carry out assessment, prediction and control tasks. Thus, the main topics in this communication are the retrieval of data from the data layer and the results of the application calculations, both to actuate over the BMS and to store these results. I-2 establishes the communication between the modules of the CLL to connect the building side and the server part of the middleware. I-3 posts events between the core communication and DAO sub-layer to the retrieval/storage of data from the BIM Server, DWH and/or external services. Similarly to I-3, I-4 enables connectivity between the core and the DAO for accessing the building data.

Communications with the data layer are established through the interfaces I-5 to I-8. I-5 implements the protocols available on the BMSs. SOAP, BACnet/IP, OPC and FTP are connectors considered within the BaaS project. I-6 makes use of Java methods provided by the BIM Server client to query this data source. I-7 defines the communication between the CLL and the DWH through mechanisms for the retrieval and storage of data. The DWH is Oracle-based, but Hibernate framework has been used with the aim of mapping the relational tables into Java objects. By its side, I-8 enables the communication with external services, i.e. weather forecast, retrieving the

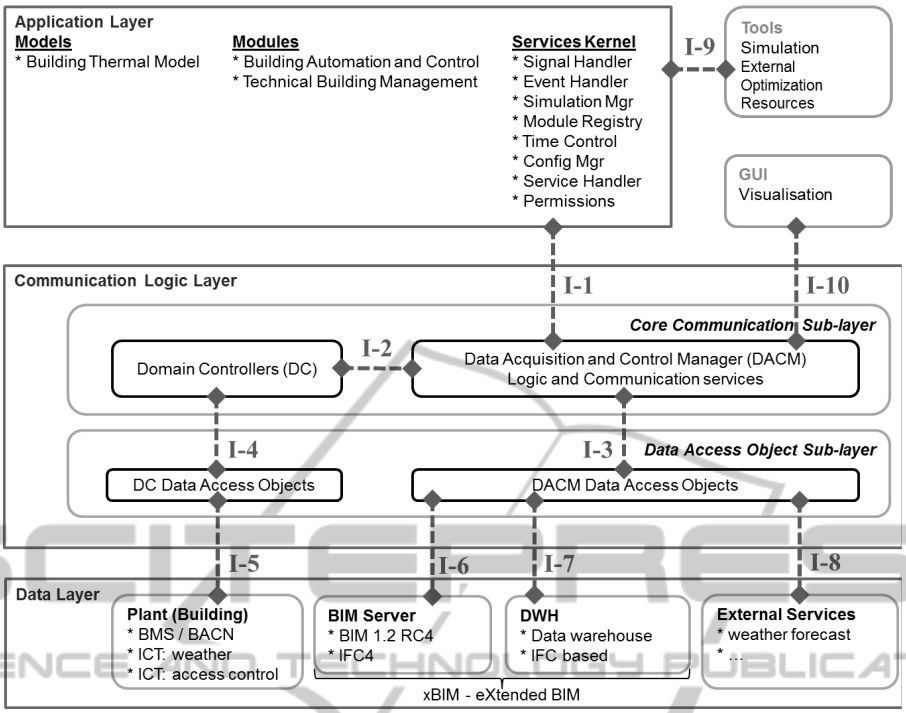

Figure 1: BaaS system high-level architecture.

information in XML format by means of HTTP calls. I-9 is out of the scope of the middleware and, finally, I-10 enables the communication from the Graphical User Interface (GUI) and the CLL to exchange information. This GUI is developed within the BaaS platform to display measurements, configure the system and access to the application results.

All the interfaces are event-driven by using the event communication mechanism provided by OSGi framework. The communication is based on three elements: Event Publisher, Event Subscriber and Event Admin. The first one represents the module that posts the events in the environment, the subscriber which listens to the subscribed events and the admin is the dispatcher of the framework.

3.2 Data Model for Data Representation

One of the greatest novelties of the middleware presented in this paper is the data-model representation for exchanging information among entities. The BaaS system objective is to perform buildings operation so as to run high-level services for the optimization of the energy performance. In this context, IFC4 (IFC, 2014) is a commonly-used standard in buildings for the representation of their information within the architecture, engineering and construction (AEC) industry. IFC4 is a large data model and comprises up to 753 objects (IFC, 2014) grouped in multiple domains.

BaaS platform is focused on the assessment and optimal control of buildings. Therefore, taking into consideration IFC4 represents the building physics, but also its facilities and systems, a reduced set of domains and classes available on (IFC, 2014) has been selected as the basic data model (Cahill et al., 2012) for the middleware to represent the information in the communication process. The useful domains in this specific system are described below.

- Structural domain: It contains the objects that represent the building physics, i.e. structure, room distribution, materials, etc.
- HVAC domain: It includes the information related to the heating, cooling and ventilation systems which are the focus of the optimal control platform and it is the basic concept for interoperability within the data model. This domain points out the equipment (e.g. boilers), terminal and flow control devices (e.g. valves).
- Building controls domain: This schema expresses the concepts about building automation, control and alarms. That is to say, devices like sensors, actuators and controllers, among others.

The shared information in the communication process is fully represented by the objects and property sets included in every domain. The relationships among domains are described by classes with

the same format commonly named *IfcRelElement*. On the other hand, the property sets define the properties of the class and how they must be interpreted. For example, a very useful object for the representation of the dynamic readings is the *TimeSeries*, described below.

```
ENTITY: IfcTimeSeries
    Name: IfcLabel
    Description: IfcText
    StartTime: IfcDateTime
    EndTime: IfcDateTime
    TimeSeriesDataType: IfcTimeSeriesDataTypeEnum
    DataOrigin: IfcDataOriginEnum
    UserDefinedDataOrigin: IfcLabel
    Unit: IfcUnit
```

The IFC4-based data model allows "context-awareness" in the middleware. Since the design phase, the components have been modeled taking it into consideration (Floeck et al., 2013). Thus, the data layer comprises sources based on IFC4, such as the BIM Server (TNO, 2010), that is IFC4 compliant. In the case of the DWH, the schema has been developed according the aforementioned domains and their property sets (Mo et al.,). Finally, the remaining components working with the data layer retrieve the data in the specific protocol and translate the information into the IFC4 common data model. For such reason, the middleware (CLL) shares, among all the entities, Java objects (Hernandez et al., 2013c) which represent the IFC4 classes in order to harmonize the heterogeneous data. Besides that, the DWH contains mapping tables with the aim of mapping the data-points identifiers from the specific communication protocol into the IFC4 context. Bearing all of this in mind, the event communication procedure shares the IFC4 objects as properties of the OSGi events under the different topics.

The advantage of using IFC4 as data model representation is the data formatting in a standard vocabulary which is widely understood within the AEC industry. Moreover, it maps the building physics and facilities through classes and attributes that eases its integration and rewards the building platform due to the "context-awareness" capability. Finally, it also enhances the scalability of the system because any additional service only needs to adapt the exchanged information into IFC4 format through the well-established interface.

3.3 Integration of Heterogeneous Data Sources

The harmonization of the data samples is the challenge in this middleware platform because, as explained, heterogeneous resources are accessed and their information has to be represented in IFC4 (Floeck et al., 2013). With the aim of interfacing them, connector components are developed and located in the DAO sub-layer of the system architecture. Each one is detailed below, but commonly all the connectors adapt the information from IFC4 to the specific data model in the data layer and vice versa.

- BMS Connector: This connector is really composed by the BMS and ICT connectors (e.g. weather station), but both are related to the building. This component is replicated for each building and its specific interface which provides the ability for the integration of additional BMSs with different protocols (Hernandez et al., 2013b).
- BIM Connector: For the assessment of the building performance, it is required to know the building facilities and systems, as well as other building features. Then, this connector reads the information from the BIM Server by means of its libraries based on IFC4 (TNO, 2010).
- DWH Connector: One of the most important components in the system, it reads/writes data from/to DWH so as to keep record of the dynamic building status, performance and configuration data.
- External Data Connector: Prediction activities in energy management systems are a necessary functionality. In that sense, this module connects to the weather forecast Web services (Hernandez et al., 2013b) to retrieve this information.

The only data-source fully compliant with IFC4 is the BIM Server, being necessary some adaptation in the remaining cases. With regard to the DWH, it extends the IFC4 data model to represent the relational tables and the use of Hibernate framework allows the representation of the information as IFC4 objects to ease the communication between CLL components and the DWH (Hernandez et al., 2013a). About the BMS and External Data connector, they require an additional adaptation though Plain Old Java Objects (POJO). The connector is able to read data and map this information into IFC4 classes represented in the POJO before posting the event (Hernandez et al., 2013c).

Finally, it is important to note that the communication is bidirectional, which implies the increase of the amount of the events in the system. For this reason, all the information associated to each connector is merged in single events (i.e. *TimeSeries* object (see above)) so as to reduce bottlenecks when processing multiple signals. In this case, another component, named General Integrator, is the responsible and it is already compliant with IFC4 (Floeck et al., 2013).

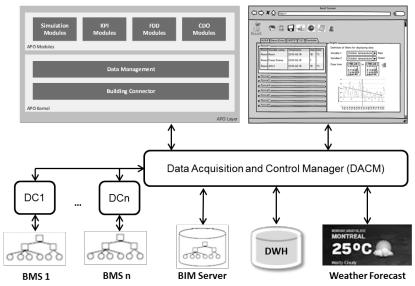

Figure 2: Deployment scheme for the BaaS system.

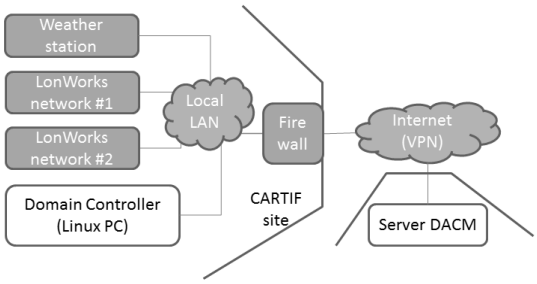

Figure 3: Deployment scheme in the CARTIF building.

4 RESULTS

By definition, a middleware is a distributed system which allows connectivity between multiple layer and entities. Then, following these premises, the deployment of the BaaS system presents an hybrid scheme, as pointed in Figure 2 (Hernandez et al., 2013a). In this scheme, there is a mixture between centralized and distributed systems following a server-client communication. The DACM illustrates the server which centralizes the communication between the building sides (i.e. BMS, DWH, BIM) and the upper layers (i.e. application and visualization). On the other hand, the client side is represented by the DCs that implement the specific communication at building level and are physically deployed in the building.

BaaS system merges four building data, therefore four DC entities that exchange information with a single DACM server through web services. The DACM can be also distributed, although the communication between upper and lower layers is still centralized in the DACM concept, if the load collapses its performance, but it is not the case. The same web service protocol is used for the communication with the application services which are deployed in an external server owing to its high performance requirements. Finally the GUI is integrated in the DACM, making up a distributed architecture that exchange information among the entities in an harmonized way.

At the current status, the deployment step in under test, although the complete communication flow is available in the CARTIF offices building located in Boecillo, Spain. In this pilot, the four connectors are deployed so as to retrieve and store data which allows keeping an historical record for further optimization tasks in the application layer. Figure 3 represents the deployment scheme in which the DACM is hosted in the server side, whereas the DC is placed in the building site. The connectivity between them is rendered by Virtual Private Networks (VPN) to avoid external invasion. Moreover, the computer hosting the DC contains the BMS and ICT connector that communicate two LonWorks-based BMS and the weather station (ICT system) and merge the information in a single event with the data translated into IFC4 to be sent to the DACM server. Finally, the DACM computer deploys the connectors with the BIM Server, DWH and weather forecast. Additional components, such as the operation schedulers, application layer and graphical user interface are being deployed in next steps.

5 CONCLUSIONS

In the current energy research topics, buildings are trend because the great possibilities they present to save energy. Mainly, there are two approaches to overcome the problem: passive solutions such as insulation system or improvement of the facilities and active ones, as for example ICT tools. This is the case of the BaaS system where a multi-layer platform has been designed for the management of the generation and distribution systems through assessment, prediction and control services.

Within this context, the middleware is a very important piece because it allows connectivity with multiple data resources which provide heterogeneous data. Each source has its own protocol and interface, therefore the middleware is able to integrate the information of the various data resources. To achieve it, integrator components merge the information in single signals so as to ease the retrieval of data to upper layers. With this aim, the middleware system bridges the gap between the Smart Grid and the information services for the analysis of operative tasks.

On the other hand, the middleware platform provides connectivity in a common format in order to avoid misunderstandings between layers. It homogenizes and shares information in the same common format, i.e. IFC4 for the BaaS specific case, through Java beans to map heterogeneous data into IFC classes. This IFC model is a widely-used standard within the building field and it includes a data schema for the

building facilities and it is also easy to integrate additional information, such as dynamic data, by means of extending the data model and its classes.

Next, taking into account the integration and coupling of heterogeneous data for sharing information in a common language, data coherency and consistency ought to be considered. In this way, the middleware implements a periodic mechanism for checking the cross-related information between the data sources. For this purpose, several components query the different data sources, compare the results and, in case of incoherences, report an alarm.

Regarding next steps and future lines, the aim is to start an operational stage by integrating all the layers in the system and run the optimal control over four demonstration buildings. Once the complete deployment will be ready, the performance tests will be carried out to detect the software goodness parameters and improve the quality of service. Finally, this development establishes the fundamentals for the replicability of the solution in multiple buildings thanks to its extensibility and scalability properties.

ACKNOWLEDGEMENTS

The authors would like to thank the partners of the BaaS project for their support during the design and development of the middleware platform. Moreover, the authors would like to thank the European Commission for the opportunity of working in this topic under the Grant Agreement no. 288409.

REFERENCES

(2010). Tno: Bimserver, open source building, http://www.bimserver.org. Last accessed 21st Aug. 2014.

(2013). In *The Future Internet*, volume 7858 of *Lecture Notes in Computer Science*.

(2014). Baas project grant agreement no.288409, https://www.baas-project.eu/. Last accessed 4th September 2014.

(2014). Campus21 project, http://campus21-project.eu/. Last accessed 3rd November 2014.

(2014). Ifc4 rc4, http://www.buildingsmart-tech.org/ifc/ifc2x4/rc4/html/. Last accessed 19th August 2014.

(2014). Netxautomation: Voyager 5.0, http://www.netxautomation.com/netx/en_voyager-visualization.html. Last accessed 5th Nov. 2014.

(2014). Smarkia: Energy efficient systems solution, http://www.smarkia.com. Last accessed 26th Nov. 2014.

Cahill, B., Menzel, K., Floeck, M., Schmidt, M., Schuelke, A., Etinski, M., Hernandez, J. L., Martin, S., and Valmaseda, C. (2012). Deliverable d3.1: High-level architecture, interfaces definitions, data models extension description. Technical Report 1.

Crosbie, T., Dawood, N., and Dawood, S. (2011). Improving the energy performance of the built environment: The potential of virtual collaborative life cycle tools. *Automation in Construction*.

Floeck, M., Schuelke, A., Schmidt, M., Hernandez, J., Martin, S., and Valmaseda, C. (2013). Tight integration of existing building automation control systems for improved energy management and resource utilization. In *Conference on Central Europe towards Sustainable Building, CESB'13*, Prague, Czech Republic.

Hernandez, J. L., Martin, S., Floeck, M., Schmidt, M., Mo, K., and Katsigarakis, K. (2013a). Deliverable d3.6: Development guide, https://www.baas-project.eu/images/deliverables/baas_wp3_d3.6_ developmentguide_1.1.pdf. Technical Report 1.

Hernandez, J. L., Martin, S., Floeck, M., Schmidt, M., Mo, K., and Rojicek, J. (2013b). Deliverable d3.3: Interfaces definition. Technical Report 1.

Hernandez, J. L., Martin, S., Floeck, M., Schmidt, M., Mo, K., Rojicek, J., and Katsigarakis, K. (2013c). Deliverable d3.5: Detailed system design. Technical Report 1.

Mo, K., Menzel, K., and Hoerster, S. Development of an ifc-compatible data warehouse for building performance analysis. In *Proceedings of the 30th CIB W78 International Conference*.

Schulke, A., Schmidt, M., and et al, M. F. (2013). A middleware platform for integrated building performance management. In Mahdavi, A. and Martens, B., editors, *Proceedings of the 2nd Central European Symposium on Building Physics*, volume 1, pages 459–467, Vienna Austria. Vienna University of Technology.

Sustainable Buildings & Climate Initiative, U.-D. (2009). *Buildings and Climate Change: Summary for decision-makers*. Sustainable United Nations, Paris CEDEX 09, France.

Zach, R., Glawischnig, S., Honisch, M., Appel, R., and Mahdavi, A. (2012). Most: An open-source, vendor and technology independent toolkit for building monitoring, data preprocessing, and visualization. In *Proceedings of ECPPM conference*, Reykjavik, Iceland.