AN EVENT-DRIVEN SOA-BASED PLATFORM FOR ENERGY-EFFICIENCY APPLICATIONS IN BUILDINGS

Cesar Valmaseda, PhD Candidate, cesval@cartif.es
Miguel Angel, Garcia, PhD Candidate, miggar@cartif.es
Jose-Luis Hernandez, PhD Candidate, josher@cartif.es
Department of Energy, Fundación CARTIF, Valladolid, Spain

Kyriakos Katsigarakis, kkatsigarakis@isc.tuc.gr
Giorgos D. Kontes, PhD Candidate, gkontes@gmail.com
Department of Production Engineering and Management, Technical University of Crete, Chania, Greece

Dimitrios V. Rovas, Professor, dimitrios.rovas@ibp.fraunhofer.de
System Integration Group, Fraunhofer Institute for Building Physics, Nuremberg, Germany and Department of Production Engineering and Management, Technical University of Crete, Chania, Greece

ABSTRACT

The topic of optimization of building operation is attracting significant interest in the community: monitoring of relevant Key Performance Indicators can help enhance state awareness and understanding; fault detection and identification can help identify irregular and ineffective operational modes; and, advanced control design techniques can yield effective/optimized operation with regards to energy performance and thermal comfort. Despite significant effort on development of algorithmic and methodological approaches to address these problems, the inherent complexity associated with practical demonstrations, has precluded testing and implementation of such approaches in realistic contexts. Within this paper, an event-driven Service-Oriented Architecture platform, is developed to address this gap and help facilitate the provision of advanced energy-efficiency and energy-management services in buildings. The use of the Industry Foundation Classes provides a standardized data-model for describing the building and its components, while the use of Model View definitions is employed to define the exchange requirements for proper software component interoperability. Data collection and homogenization from the building is addressed through an abstraction layer, capable of hiding many of the intricacies and providing a clean interface for the development of building services. An exemplary application of the proposed architecture in a real office building in Greece is presented.

Keywords: BIM and its applications, IFC, MVD, energy management, fault detection, control design

1. INTRODUCTION

Buildings are contributing significantly to global energy consumption: it is estimated that 70% of electricity and 40% total energy use can be directly attributed to the building sector. Tools to deliver energy-aware monitoring and optimized operation services can help reduce part of the energy demands and can yield significant energy and economic benefits. Development of platforms for hosting such energy-related building services should as a requirement functionally support the following three distinct groups of services: Fault Detection and Diagnostics, containing services to detect malfunctioning or performance-degradation of building components and, where possible, capable of identifying the root cause so that maintenance can be scheduled; Control Design and Optimization services, to design and apply intelligent building control strategies; and Key Performance Indicators

(KPIs) calculation to monitor pertinent energy- and comfort-related performance parameters, thus increasing user energy awareness and contributing to enhanced monitoring. These groups of services – continuously recurring during the building operational phase – will be collectively referred henceforth as APO services: (A)ssessment of the current building state; (P)rediction of the effects that various decisions will have to KPIs; and (O)ptimization of performance as measured through relevant KPIs.

While significant effort has been focused on developing algorithms for such services – see for example (Kontes et al. 2012, Ma et al. 2010 and Olderwurtel et al. 2012) for control design optimization techniques and (Kukal et al. 2009 and Trojanova et al. 2009) for Fault Detection and Diagnosis approaches, – implementation of such approaches in realistic contexts is lagging due to increased complexity, thus the proper design and viability of such a hosting platform is essential.

Transparency, openness and interoperability are crucial design requirements. In this effort, the service composition framework OSGi (OSGI Alliance 2013), a proven and freely available technology to provide software platform is successfully demonstrated in several works: in (Jahn et al. 2012) to facilitate the communication of heterogeneous BMSs and devices networks in buildings; in (Floeck et al. 2013a) to allow the integration of existing ICT systems for performance evaluation, benchmarking, and load balancing based control; the Open Gateway Energy Management Alliance (OGEMA 2013) has developed components to adapt the consumption according to the generation in a power system where fluctuating renewable resources are incorporated; or, in (Zach et al. 2012) tools for building data collection and pre-processing tasks have been developed. However, these software environments based on service-oriented architectures (SOA) could be enhanced when combined with an event-driven architecture (EDA), where state changes of the system trigger the generation of events and any other service or component which has subscribed to the event will then consume the event and be able to react to it.

The goal of the present paper is to present such a design developed in the context of the FP7 EU ICT Project Building as a Service (BaaS). The BaaS platform has a modular event-driven design based on SOA and EDA principles, capable of facilitating the provision of advanced energy-efficiency and energy-management services in buildings, while meeting a set of transparency, openness and interoperability requirements.

Moving forward, the aforementioned efforts still lack the full interoperability concept necessary to became useful the building information along the building lifecycle. In (Crosbi et al 2012, Böhms et al 2011) an open source platform for the semantic integration of energy information in buildings is presented, highlighting the benefits of creating common sets of vocabularies along the building lifecycle. The present work revisits this topic upon the utilization of Industry Foundation Classes (IFC4) standard (ISO 16739 2013), for describing the building and its constituents, and can be used for the parameterization and initialization of the platform or its services. The use of IFC Model View Definitions (MVDs) provides a convenient mechanism for enforcing interoperability and satisfying exchange requirements between various components of the platform.

2. THE BAAS PLATFORM ARCHITECTURE

The definition of the overall platform architecture is guided by three design principles: use of open technologies, integration of Service-Oriented Architectures (SOA) and Event-Driven Architectures (EDA), and cloud-based distributed functional design as explained below and in (Floeck et al. 2013b, Floeck et al. 2013c). Shown in Figure 1 is a high level view of the design of the BaaS platform.

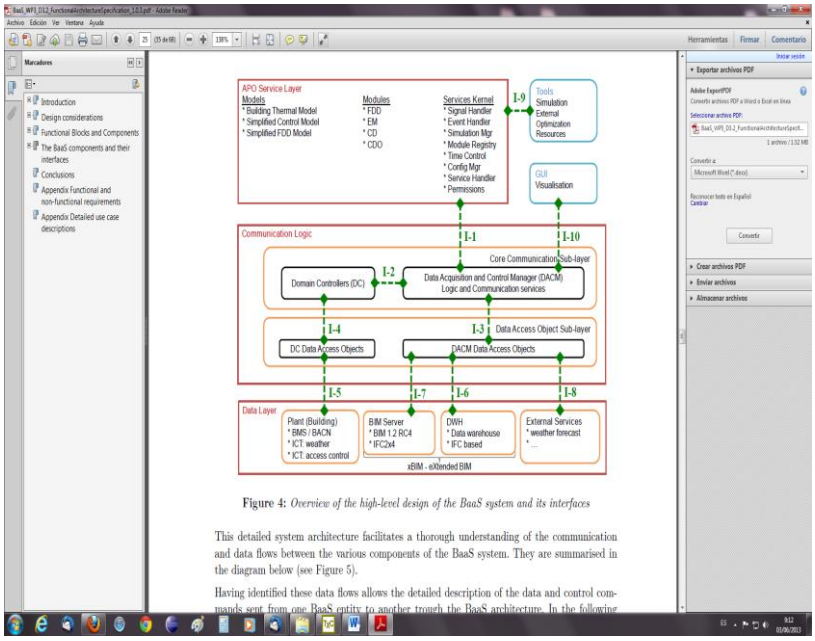

Figure 4: *Overview of the high-level design of the BaaS system and its interfaces*

This detailed system architecture facilitates a thorough understanding of the communication and data flows between the various components of the BaaS system. They are summarised in the diagram below (see Figure 5).

Having identified these data flows allows the detailed description of the data and control commands sent from one BaaS entity to another trough the BaaS architecture. In the following

Figure 1 High-level design of the BaaS platform and communication interfaces

A three layer architecture has been selected: the Data Layer (DL) serving static and dynamic data through a set of data sources or repositories (a data warehouse, a BIM server, a Building Management System (BMS)/Building Automation and Control Network (BACN), and external services); the APO Service Layer (APO SL), providing support for hosting the reasoning and analytics services; and the Communication Logic layer (Middleware Layer) acting as an abstraction layer to facilitate communication between the Data Layer and the APO Services Layer. These functionally disjoint layers operate independently, communicating through the use of properly-defined (software) interfaces, denoted as I-1 to I-10 in the Figure above.

A guiding principle in the design and implementation of the BaaS platform has been the use of open middleware technologies to enable interactive cloud environment: the service composition framework OSGi is a proven and freely available technology for middleware implementations, that has been used for the development of the BaaS platform. The use of open FLOSS (free/libre open source software) is promoted for BIM Querying (BIM Server) or accessing data. A second aspect of importance has been the integration of SOA and EDA architectures. In modern enterprise architectures, business processes are very often mapped to IT services. Ideally, these services are self-contained, re-usable, and accessible via a well-defined interface. The services can then provide their functionality to any entity that requires them. Typical software environments which implement this design paradigm are service-oriented architectures (SOA). When combined with an event-driven architecture (EDA), state changes of the system trigger the generation of events which in turn are broadcast. Any other service or component which has subscribed to the event will then consume the event and be able to react to it. Through, the use of the SOA and EDA software design paradigms leads to self-contained services which can be seen as independent entities. By design, they are only loosely coupled and because of the communication mechanism between the components and services which makes heavy use of events renders the platform well-suited for distributed (cloud) deployment. A more detailed description on the functionality and design of the three layers is given in the sections below.

2.1 Data Layer

The layer functionally supports the collection, processing, and monitoring of building performance, geometric, parametric, operational and environmental data. The Data Layer, shown in Figure 1, comprises four components: *Building Management System (BMS), Data Warehouse (DW), Building Information Model (BIM) and External Data (ED)*:

- The Building Management System (BMS) is central to any building control solution. BMSs act as information gateways for communicating information from/to building sensing and actuation modalities. Moreover they implement the low-level logic for system control (HVAC, lighting) along with supervisory control approaches. If SCADA systems are installed as part of the BMS the data can be stored in a database and be made available to building operators for further processing.
- The Data Warehouse is a tool which supports the management and processing of granular bulk data. In typical BMS/SCADA installations a simple data-base is used; instead, the use of a DW allows categorizing and analyzing this sensed (fact) data in combination with other descriptive data. The additional benefits of using a DW is the possibility of post-processing information so that relevant to the end-user Key Performance Indicators (KPIs) can be computed. Users and applications (e.g. APO services) can directly retrieve and use these pre-computed values. In addition, the BaaS data warehouse schema has been developed in conformance with the ISO 16739:2013 standard (IFC4) (Flynn et al. 2012).
- BIM Information is managed centrally through the use of a BIM server. The server hosts all IFC related data and queries to the data model can be performed so that building information can be obtained as needed. The desire for openness has led to the use of the open-source TNO BIMServer, that was properly adapted to support the IFC4 schema. The BIM model can be queried to acquire: (1) aggregated geometrical information describing rooms, zones, the thermal mass of walls, the size of glazed areas, the thermal resistance of these elements, etc. (2) the topology of building services systems but also the topology of architectural, space related components (e.g. which rooms belong to a thermal zone, which thermal zones belong to which floor, etc.). (3) the relationships between system elements (e.g. spatial containment of sensors).
- External Data Services: APO services needs to consider data from sources relevant to the building but external to the BMS. Such data may include weather station data and occupant density data (or occupant profiles from security systems), or even prediction data like weather forecast (from weather forecast providers) or occupancy forecasts (from occupancy scheduling systems).

2.2 Communication Logic Layer

The Communication Logic Layer (CL), acting as a middleware, links the Data Layer and the APO Service Layer in a bi-directional manner: data from various sources (BMS, BIM, data warehouse, etc.) making up the Data Layer can be requested by the APO services while at the same time data can be sent from the APO services to the Data Layer, e.g. optimization results to be stored in the data warehouse or HVAC set-points sent to the BMS.

The CL itself is divided into two sub-layers: the Core Communication Sub-layer (CC) and the Data Access Object Sub-layer (DAO). The CC Sub-layer is composed of two different modules, i.e. the Domain Controller(s) (DC) and the Data Acquisition and Control Manager (DACM). The Domain Controller runs on the site of the asset being monitored. It provides low-level access to the physical asset, e.g. by connecting to the BMS for data retrieval or actuation triggering, local data buffering, or BMS discovery (i.e. scanning for sensors and actuators). In case of very large or distributed assets (e.g. large industrial plants or operating sites of a company spread over a large geographical area), more than one DC can be present. That means that multiple DCs can be managed by a single DACM to monitor and control one asset or that multiple assets can be controlled simultaneously by a single DACM or a combination thereof.

In contrast, the Data Acquisition and Control Manager is not required to, but can, run on-site; it is more likely that the DACM runs in the controlled environment of a managed data center. The DACM is linked to the DC(s) and provides high-level functionality, e.g. connectors to components outside the CL, such as APO services or external cloud-based data sources (weather data, etc.). In the same way that one DACM can control multiple DCs, a plenitude of APO services can be connected to a DACM. While the APO services provide advanced and computationally intensive functionalities, the DACM offers interfaces to other components of the DL, e.g. the data warehouse, the building infrastructure model (BIM), or external data services. Moreover, the DACM provides generic scheduling and data handling functionalities to the APO layer. The DAO sublayer is responsible for the identification and communication to the components of the Data Layer, that is, where the interfaces are

implemented. Any future extensions to internal or external data sources could be managed in the DAO Sub-Layer, allowing flexibility to integrate new information services.

The MW functionality is provided by a number of functional blocks implemented as one or more software bundles. The Manager block is in charge of the management of the system configuration parameters, such as users, KPIs, alarms, security, and schedulers. It receives the requests from a user interface and its operation is the configuration of the system. It is also in charge of managing the calculations, privileges, etc. of the CL. The APO Layer Connector block implements the communication of the CL and the APO SL via the interface I-1. This interface is exposed by the DACM to the APO Service Layer. The APO-SL in turn implements this interface so that the APO kernel (described in the next section) can access it. The APO Layer Connector retrieves the data needed for the optimization and simulation tasks and reads the results in order to actuate in the BMS. It also needs to be kept in mind that the APO Service Layer may be located in another environment or domain than the CL and may thus be beyond the control of the CL. In this case, the APO-SL will connect to the CL via an IP network. To ensure that those connection requests can be safely accepted, best practices with regard to the implementation of trust models must be followed (e.g. applying access control and access permissions). In the case of the BaaS system, this means that any component requesting data from the CL must properly authenticate itself in order to be granted access and that – depending on the permissions associated with the authentication credentials – access permissions will only allow access to those files which the component actually needs. Finally, the data layer connector block implements the functionality to permit communication between the CL and the data layer. However there are several entities in the data which need the communication with the CL. Thus, this global functional block is divided into several specific ones. Each one is in charge of the communication with the appropriate entity. Similarly to the case of the APO Layer Connector, the Data Layer Connector uses the same authentication and authorization processes to ensure that only trusted connections are established.

2.3 APO service layer

The APO Service Layer is in charge of carrying out all aspects of building operation, it is the host of the business intelligence of the overall platform. APO Services are hosted in this layer and the APO Kernel is an interoperability component for the management of these services and for catering to data access needs. This set of services can be classified according to their functional role in three groups :

- Fault Detection and Diagnostics (FDD), containing services that provide analytics to detect and identify the cause of malfunctioning components of the building. This includes Energy management (EM) services, that provide analytics for monitoring the performance of building components at various hierarchy levels (from the whole building to specific equipment) and notify the administrator on performance degrading equipment.
- Control Design and Optimization (CDO), containing services that provide control-related tasks optimizing the control strategies applied to the building.
- Key Performance Indicators (KPIs), containing services that provide analytics for monitoring several cost- energy- or comfort-related performance indexes for the building.

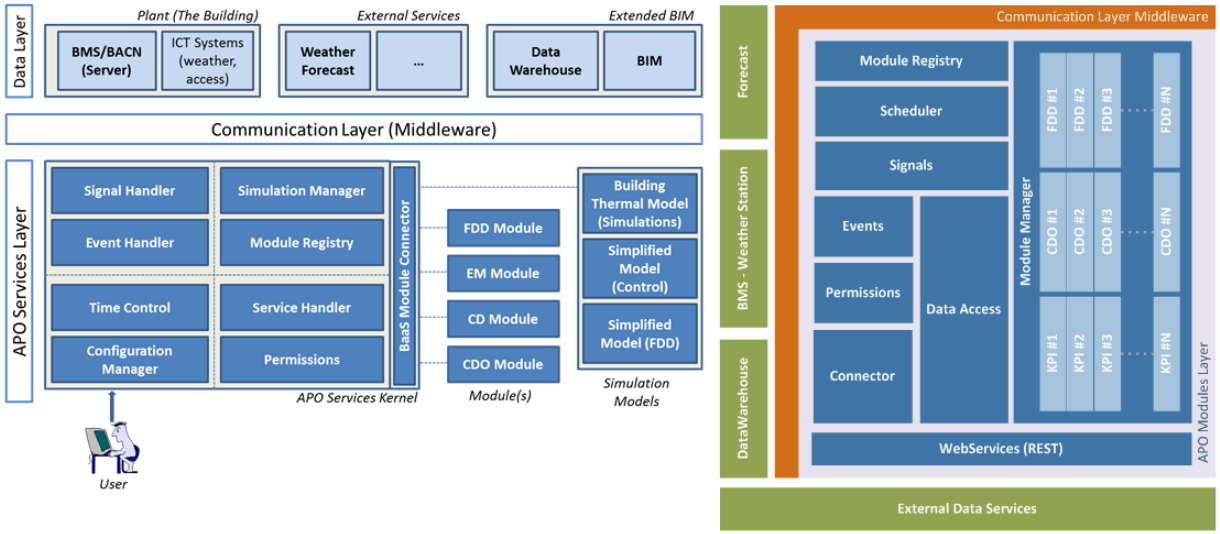

Figure 2 APO Services Layer – Core components (Left); and Software Components (Right)

A schematic representation of the APO kernel, containing all the necessary functional components, is shown in Figure 2 (left). During the initial deployment of the system, the administrator registers a set of modules in the module registry (visible in APO modules layer in Figure 2 (right)), and are made available for their invocation. The APO Kernel Scheduler is in charge of invoking these models in response to events, or on user-defined intervals. The modules can implement small functionalities (e.g. a KPI Calculation) or more complex ones (e.g. model-based predictive control design). During the operational phase of the system new modules can be subscribed to the module registry, since all kernel components are implemented as bundles of an OSGi framework. This enables the scalability and interoperability of the system, since all kernel components can be installed, updated and removed from the framework, without influencing the building operation or the other services, thus providing significant flexibility. A number of data abstractions have been defined: the signal component provides direct access to current measurements (obtained from the BMS), the schedule represents an abstraction for time-series data which are obtained, typically, from the DW; and specific objects allow querying the current BIM model of the building

Subsequently, the configuration manager is responsible for adjusting the available APO modules to the target building, using two available interfaces: user input and/or the BIM repository. For the user input, the building manager or the installation engineers parameterize the APO modules to be applied to the building through simple GUI. Here, the inputs for each analytic, the execution intervals an analytic re-initiates, the event that triggers an analytic, etc. are determined for the whole building.

3. DATA REQUIREMENTS – BIM-AWARE CONFIGURATION

3.1 Data Requirements

The platform described above helps through the abstraction layers and the separation of components to collect and make available information to APO modules that implement the business logic. A significant challenge is the configuration of the APO analytics and the platform itself which can be a time-consuming and error-prone process. Within BaaS this configuration is performed (semi-)automatically by querying for information from the BIM model of the building. In this approach, during the configuration phase, installation of an APO module (which is delivered as a fragment project) invokes a configuration class. Within this class, queries are performed to identify relevant information. To make the discussion more concrete consider a simple example, that of a KPI calculation module that, at the end of each day, computes the average temperature of each office during the working hours; here the APO module contains the information regulating the initiation of the calculation (e.g. every day at 23:00), the interval used for the calculation (e.g. 08:00 – 16:00) and the information that the

averaging process is targeted to an IfcSensor object (BuildingSMART Alliance 2013a) of TEMPERATURE IfcTypeEnum (BuildingSMART Alliance 2013b). Subsequently, a query to the BIM repository discovers the temperature sensors in each room and provides to the configuration manager a table mapping rooms to sensor ids. Finally, the configuration manager instantiates the necessary objects of the APO module (one for each temperature sensor - room pair) and deploys the analytics, without requiring user input.

While the necessary BIM data for configuring the FDD and CDO modules are most often limited to the BMS components and HVAC systems of the building, it is very often that a simulation model can be invoked to perform, say, a complex KPI calculation. In this case, the information for setting up the simulation model can be inherently more complex (Bazjanac 2007). It is very often that for simulation purposes geometric information is needed, and in particular a challenging task is the identification of 2nd-level boundaries. When this information is not already included in the BIM model, algorithmic for semi-automatic identification of these boundaries has been developed and can be used to pre-process data so that they are available for simulation purposes – giving more details is beyond the scope of the present paper and related results are reported elsewhere (Lilis and Rovas 2013).

In general, simulation input data can be classified with respect to the frequency of their change into:

- Static Data (SD), which include the building geometry, construction materials, glazing information, systems used in the building, etc.;
- Dynamic Data (DD), which consist of all time-dependent data, like user-actions (e.g. opening and closing the windows), occupancy schedules in each of the building zones, use of equipment, weather predictions, BEMS commands, etc.

Static data are relatively well-known and can be obtained from the BIM model. In the case of use of simulation during the operation phase, sensed data (e.g. occupancy patterns, or external weather conditions) can be used to improve the quality of predictions of the simulation models. In our case, the dynamic data are aggregated from various in-building and external sources and are recorded to the DW. Then setting up a simulation task, involves accessing these data directly from the DW, and passing them to the simulation engine using a co-simulation setup (Kontes et al. 2012). The integration provided by the BaaS platform allows for transparent inclusion of both static and dynamic data and their use for simulation purposes – a more active role and use of simulation models during the building operational phase is therefore advocated.

3.2 BIM Repository

To ensure the viability and adoptability of the BIM repository within the proposed platform, the TNO BIMServer has been adopted as the BIM repository of BaaS. The TNO BIMServer (BIMServer 2013). is an open-source platform, enabling full access to the code and offering customization capabilities. The latter is also enabled by the modular architecture of the server, based on plugin development; here the vital tasks supported by the server are developed as plugins available to the user. This way, more than one plugins for the same task can be available, addressing different requirements, while vendors are able to develop custom solutions suitable for their needs and incorporate them into the server through the plugin interface, without modifying the server core.

Moving forward to the data management requirements, TNO BIMServer utilizes the Eclipse Modeling Framework (EMF) to provide a modular and object-oriented representation of the IFC schema. Here, the IFC STEP/EXPRESS file is parsed and the included classes (more than one thousand) and their interrelationships are converted to an EMF Core (ECore) file. Subsequently, the EMF framework extracts the information from this file and generates Java classes that are passed to the database layer. The database layer stores the entries in key-value pairs accessible through the KeyValueStore Interface. The implementation of the database is Oracle Berkeley DB Java Edition, an open source, embeddable, transactional storage engine written entirely in Java and running in the Java Virtual Machine without the need of a remote server. In difference to relational database, this implementation stores object graphs, objects in collections, or simple binary key/value data directly in a B-tree on disk. Using this approach, functionality similar to version control software (like subversion) is provided and the versioning system is governed by the following principles:

- Models are stored in projects, while a new model version is a new project revision.
- All project revisions are accessible and cannot be altered

This way, any model update (full or partial) is supported as a new project revision, while the availability of the revision history allows for reverting to previous versions automatically. Within the APO Kernel a BIM manager is in charge of storing a local copy of the model. The BIM component monitors the Server for changes to the model. When such a change occurs, the local copy of the model is automatically updated by contacting the server using a web-service interface. This way always a current version of the BIM model exists locally, and is made available for querying directly to all the components. Note here, that in order for the repository to be up-to-date with the current state-of-the-art technologies, IFC4 support is essential. Also support for describing sensor and actuator objects is only available in IFC4. Due to the transitional phase from IFC2X3 to IFC4, all available server versions support IFC2X3, thus a version of the TNO BIMServer (1.2 nightly build – 26/09/2012) was adapted to support the IFC4 standard.

3.3 Model View Definitions

Although the utilization of the BIM repository and the IFC4 data model allows for semi-automatic deployment and operation of APO services for each target building, at the same time inserts more complexity to the problem rather than simplifying the task, since it requires by all software modules to support the IFC data model. The IFC4 can be sufficiently rich or lean depending on the editor and the amount of information present. In order to define the specific IFC subset each APO service group (CDO, FDD, EM, KPI) has to support, the concept of MVDs has been adopted.

According to BuildingSMART alliance (http://www.buildingsmart.org/standards/mvd), an MVD "defines a legal subset of the IFC complete schema and provides implementation guidance for all IFC concepts (classes, attributes, relationships, property sets, quantity definitions, etc.) used within this subset. It thereby represents the software requirement specification for the implementation of an IFC interface to satisfy the Exchange Requirements".

Following this approach, the Exchange Requirements (ER) of each type of APO services have been defined, documented and communicated. Subsequently, these requirements were described using the MVD formulation and are made available to all vendors and service providers. Within this context, if two software components are to interact they need to exchange sufficient information – all the exchange requirements so that this communication is complete are defined in the MVD, while the BIM repository acts as the aggregator of such information, and the provider to clients (via available interfaces) of the requested information. Moreover, the availability of the BIM and the MVD description allows the generation of queries to the BIM based on the MVD, since the MVD actually determines which queries are supported, i.e. services can expect some meaningful data in the response.

4. EXAMPLE

4.1 MVD necessity example

The necessity of MVDs will become obvious through a use-case scenario. Consider the office building shown in Figure 3, housing the Technical Maintenance Services of Technical University of Crete, located in Greece. Each office is equipped with an AC unit for cooling and includes the following sensors: a temperature sensor; a humidity sensor; a contact sensor mounted in each window and door; and an energy meter for each split-type AC.

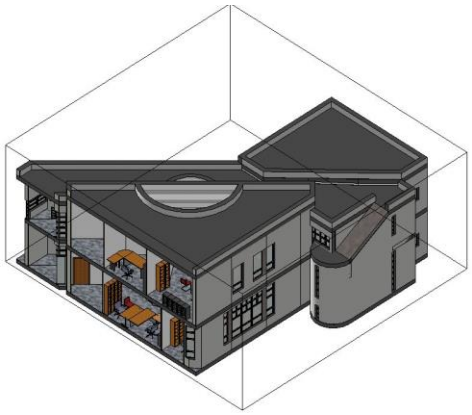

Figure 3 Overview of the TUC test building

The information on which contact sensor belongs to each room is vital for the APO services; for example for an FDD module that has to ensure that the AC of a room is not operating when a window is open. To acquire this information, the IFC4 model of the building (designed in Constructivity) is uploaded to the IFC4-enabled TNO BIMServer. During the design process, contact sensors are attached to each window/door element they monitor, while all other sensors were "attached" (using an IfcRelAggregates relationship) to the respective IfcRoom entity they belong to. From there, using the querying functionality, this information can be requested.

An initial query is performed, supporting the following steps: a) get all IfcRoom objects of the building; and b) populate the IfcSensor objects that belong to each IfcRoom. Upon completion, this query manages to discover only the energy meters and the temperature, humidity and luminance sensors, but fails to identify that the building is equipped with contact sensors, since they are not attached to the respective IfcRoom entities. To overcome this problem, a different query is designed and deployed that: a) get all IfcSensor objects; and b) discover in which IFC object they are assigned to. This query correctly discovers all building sensors, and a new query is required to determine in which IfcRoom each window equipped with a contact sensor belongs to. Moreover, a potential problem arises if an IfcSensor object is not attached to any room element (for example a sensor is attached to the roof of the building).

In order to provide a generic solution to the specific problem, an MVD is defined, requiring all IfcSensor objects to be assigned to an object they serve (IfcRoom, IfcWindow or other). The impact of the specific MVD rule to the designed IFC files is twofold: not only each IfcSensor object should have an additional relationship information to the allow for spatial or system containment, but it also requires that these related elements are defined. Under this rule this information is required for the second querying approach described above to be meaningful. If a model is provided which is not compliant with the MVD, checking these set of rules will force the designer to include this information as it is functionally useful for the type of services to be provided. This simple example, hopefully serves to identify the undisputed value and necessity of the MVD in the process. Obviously supporting complex APO services requires a more elaborate and detailed set of rules – this is part of ongoing work and a concrete MVD definition is to be presented in a future paper.

4.2 Example analytic

Having the MVD-compliant building model at hand, allows for automatically configuring a plethora of APO services. For the present work, let's consider the example of an FDD analytic that checks if the AC of a room operates while a door or a window is open. Here, an APO module is designed, containing the following structure:

- Inputs: a contact sensor id and an energy meter id.
- Output: True, if the contact sensor indicates an open window.
- Activation event: the analytic should initiate when the energy meter of an AC unit indicates operation.

Once the specific model has been registered to the module registry, a targeted query to the BIM Server is initiated, to identify all the possible input pairs (i.e. energy meter id – contact sensor id), to instantiate the proper number of APO services that implement the analytic for the whole building. Figure 4 (left) shows the result of the

specific query for office_9 (O9_x entries); here three instantiations of the analytic are implemented, since the room is equipped with two contact sensors in the windows and one contact sensor mounted to the door, while Figure 4 (right) shows the automatic input configuration of each module. When the energy meter serving the AC of office_9 triggers the pre-defined event (i.e. the AC is operating), the three analytics are invoked and, when a contact sensor in office_9 is active, generate a fault detection event that can be further processed.

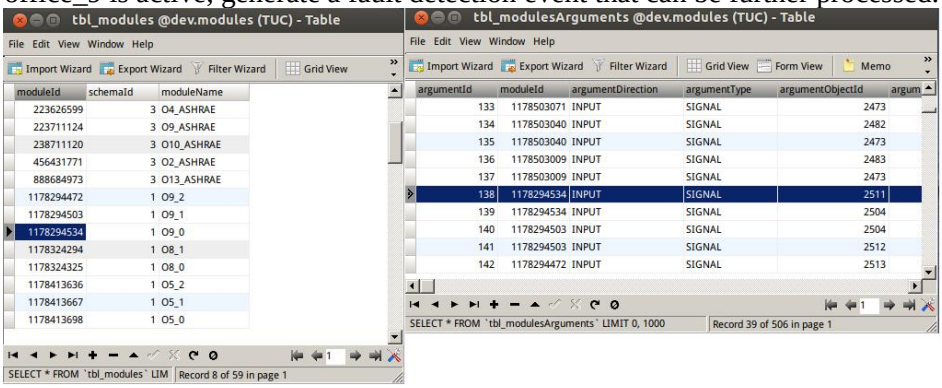

Figure 4: Energy meter - contact sensor pair ids in each room of the test building, as a result of a query to the TNO BIMServer

5. CONCLUSIONS

In the work presented here, the development of an event-driven Service-Oriented Architecture based platform capable of facilitating the provision of advanced energy-efficiency and energy-management services in buildings is addressed. The necessity for transparent data management and interoperability is ensured by the use of the IFC standard, accompanied by proper MVD definition for the exchange requirements and the utilization of a suitable BIM server. Technical viability of the proposed approach is showed by means of a proof of concept based on a semi-automatic APO service configuration and deployment process, for an office building, located in Greece. As part of ongoing work this solution is to be deployed in five demonstration buildings (three office buildings, one hotel and one school) to test and evaluate the potential for aggregating information.

ACKNOWLEDGEMENTS

The research leading to these results has been partially funded by the European Commission FP7-ICT-2011-6, ICT Systems for Energy Efficiency under contract #288409 (BaaS).

REFERENCES

Bazjanac, V. (2007). "Impact of the US national building information model standard (NBIMS) on building energy performance simulation". *Presentation at the Building Simulation 2007 conference, Beijing.*

Bimserver (2013), https://code.google.com/p/bimserver/wiki

Böhms, M. and Van den Helm, P. (2011). "The IntUBE (Energy Information) Integration Platform". *2nd Workshop on eeBuildings Data Models, Proc. of CIB W078 & W102 conferences, Sophia Antipolis, France, 2011.*

BuildingSMART Alliance (2013a). "IFC4 documentation". http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm (30/05/2013)

BuildingSMART Alliance (2013b). "IfcSensorTypeEnum". http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcbuildingcontrolsdomain/lexical/ifcsensortypeenum.htm (30/05/2013)

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, vol. 20, no. 2, pp 205–216.*

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N. and Weerawarana, S. (2002). "Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI". *Internet Computing, IEEE, 2002, 6(2), 86-9.*

Floeck, M., Schuelke, A., Schmidt, M., Etinski, M., Valmaseda, C., Hernández, J.L. and Garcia, R. (2013a). "Reducing the energy footprint of infrastructure buildings by unlocking synergies between available building automation, energy management, and building information modelling systems". *Proc. of ACRECONF 2013, New Delhi, India, 2013.*

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

Floeck, M. Santos, N., Schmidt, M., Schülke, A., Hernández, J.L., Martín, S., Valmaseda, C., Rovas, D.V. and Rojíček, J. (2013c). "Functional Architecture Specification". *BAAS Deliverable D3.2, 2013*

Flynn, D., Menzel, K., Cahil, B., Schmid, M., Schulke, A., Rovas, D.V., Valmaseda, C. and Martin, S. (2012). "Data Warehouse Requirements and extended BIM specification". *BAAS Deliverable D2.1.*

ISO 16739 (2013). "Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries". *The International Organization for Standardization*.

Jahn, M. et al. (2012). "Towards a Context Control Model for Simulation and Optimization of Energy Performance in Buildings". *3rd Workshop on eeBuildings Data Models, Proc. of ECPPM conference, Reykjavik, Iceland, 2012.*

Kontes, G.D., Giannakis, G.I., Kosmatopoulos, E.B. and Rovas, D.V. (2012). "Adaptive-fine tuning of building energy management systems using co-simulation". *2012 IEEE International Conference on Control Applications (CCA). pp. 1664-1669, IEEE.*

Kukal, J., Macek, K., Rojicek, J. and Trojanova, J. (2009). "From Symptoms to Faults: Temporal Reasoning Methods". *International Conference on Adaptive and Intelligent Systems (ICAIS'09), pp. 155-159, IEEE.*

Lilis, G.N. and Rovas, D.V. (2013). "CBIP: Common-Boundary-Intersection-Projection". *In preparation.*

Ma, Y., Borrelli; F., Hencey, B., Coffey; B., Bengea, S. and Haves, P. (2010). "Model predictive control for the operation of building cooling systems". *American Control Conference (ACC). pp. 5106-5011.*

OGEMA (2013). http://www.ogema.org/ [online, last accessed 12 Jun. 2013]

Oldewurtel, F., Parisio, A, Jones, C.N., Gyalistras, D., Gwerder, M., Stauch, V., Lehmann B. and Morari, M. (2012). "Use of model predictive control and weather forecasts for energy efficient building climate control". *Energy and Buildings, 45, pp. 15-27.*

OSGi Alliance (2013). http://www.osgi.org. [online, last accessed 12 Jun. 2013].

Trojanova, J., Vass, J., Macek, K., Rojicek, J. and Stluka, P. (2009). "Fault Diagnosis of Air Handling Units". *Fault Detection, Supervision and Safety of Technical Processes, pp. 366-371.*

Zach, R., et al. (2012). "MOST: An open-source, vendor and technology independent toolkit for building monitoring, data preprocessing, and visualization". *Proc. of ECPPM conference, Reykjavik, Iceland, 2012.*