Functional and technological definition of BIM-aware services to assess, predict and optimize energy performance of buildings

J. Rojicek & R. Fisera
Honeywell Labs, Prague, Czech Republic

G.D. Kontes, G.I. Giannakis, G.N. Lilis & D.V. Rovas
Department of Production Engineering and Management, Technical University of Crete, Chania, Greece

ABSTRACT: There are a number of important elements in designing of building energy management systems – when data collection, aggregation and management is usually well addressed by existing building management systems, actual analytical components allowing to diagnose energy-prone and/or user comfort compromising behaviors are far less mature. It is not only about developing algorithms for such tools, but also proper design of a hosting platform and its viability – it should not only enable access to sensor readings, but also provide access to other building data like Building Information Models and allow collaboration and interconnection of such analytics. BaaS project calls such tools Assess, Predict and Optimize services. Developing a smart platform supporting these services naturally leads to a concept of the building as a service ecosystem (BaaS) where any new tool can be plugged in the system and can benefit from already existing components. In the present work, the high level architecture of the BaaS platform is presented and the ability of BaaS system to act as a platform enabling the building contextual data as well as dynamic data (sensor readings) to software modules is demonstrated through a use-case example on a simple one zone office building.

1 INTRODUCTION

Buildings are important contributors to the total energy consumption, thus Building Energy Management Systems (BEMS) are key ingredients towards enabling parsimonious use of energy resources. Existing BEMS, although facilitating near-complete mechanisms for data collection, aggregation and management, lack the analytical components allowing diagnosing a behavior leading to excessive energy consumption and/or compromised occupants' comfort. BaaS project calls such tools Assess, Predict and Optimize (APO) services.

Even though research effort has been focused on developing algorithms for such tools showing significant results (e.g. see PEBBLE Project, OptiControl Project), proper design of the hosting platform and its viability are essential. Such platform should support access to Building Information Models (BIM), thus providing a common interface between the various analytics and enable collaboration and intercommunication between them in a transparent way.

Several research projects aiming to provide such a holistic approach are on the way: the Monitoring System Toolkit (MOST Project) is a set of tools enabling effortless measurement, processing and visualization of in-building data streams; Control and Automation Management of Buildings and Public Spaces (CAMPUS21 Project) develops a Hardware-Software-Platform for the integration of existing ICT-subsystems supporting energy, building, and security systems management for energy-efficient operation of public buildings and spaces; and ICT Platform for Holistic Energy Efficiency Simulation and Lifecycle Management of Public Use Facilities (HESMOS Project) provides advanced simulation capabilities to decision makers and attempts to close the gap between BEMS and BIM in an effort to outcome energy- and cost-minimizing decisions throughout the whole building life-cycle.

Within BaaS Project, a smart platform supporting the whole ensemble of APO services (i.e. Control Design Optimization, Fault Identification and Energy Benchmarking) is developed, where each new service can be plugged into the platform and benefit from already existing components, thus leading to a concept of the building as a service ecosystem (BaaS). This also implies "Software as a Service" (SaaS) marketing model.

Such solution not only improves system performance by detecting and correcting inefficiencies, but it also increases user awareness: having all relevant Key Performance Indicators (KPIs) in one place it is easier to monitor discrepancies from their expected values or any other KPI deterioration. KPIs usually cover energy consumption, occupants' comfort and ecological friendliness of the system. Even if KPIs evaluation cannot be done automatically, or if the system control cannot be altered manually (e.g. for safety reasons) the man in the loop has all necessary information at his hands, so it is easier to make qualified decisions.

In the present work, the high level architecture of the BaaS kernel hosting the APO services is presented and the ability of BaaS system to act as a platform enabling the building contextual data as well as dynamic data (sensor readings) to software modules is demonstrated through an illustrative example on a simple one zone office building.

2 STATE OF THE ART OF APO SERVICES

APO services address three main areas: Fault Detection and Diagnostics (FDD); Energy Management (EM) and Control Design and Optimization (CDO). Though they are treated separately, one can clearly see that they can be closely related.

Recent commercially available FDD services are typically provided in form of SaaS. The algorithms running on the cloud are typically designed based on accumulated expert knowledge and are searching for unusual patterns in the data collected from building sensors. As the contextual information for the target commercial buildings is available usually only in very limited form and with quality varying from site to site, the FDD solution, aspiring to wide applicability without complex settings required, needs to be sufficiently robust. This implies that the algorithms should be capable of producing accurate results even with hardly any contextual information available. The lack of context can, in some cases, limit the fault detection and mainly diagnostics capabilities of the service. Basically, there are two main approaches in the FDD research field: rule-based (Kukal et al. 2009, Schein et al. 2006) and model-based (e.g. Du and Liang 2007) (and combinations). Typically rule-based methods are continuously evaluating simple rules, or sets of rules, using sensor data collected from the building, while a fault reasoning process subsequently decides on particular fault's presence. Model-based methods typically compare the measured value of proper KPI against the modelled one. These referential values are constructed exploiting various approaches ranging from black-box, when hardly any context information is available, to white-box, when the monitored equipment is known in detail. Often only one piece of equipment is focused separately neglecting the fact, that it is commonly a part of a complex building system, where faults diagnostics is a difficult task (fault masking & propagation) hardly solvable without having a contextual information at disposal (e.g. building equipment connectivity model). The availability of building context (BIM) can thus significantly improve the performance of fault detection and mainly fault diagnostics algorithms.

Moving forward, energy management services are often a part of BEMS. Typically the energy consumption on the whole building level and several major energy consumers are monitored. More advanced systems provide baselining or benchmarking functionality, usually on the building envelope level. In the former case, the referential energy consumption is constructed from past data (e.g. searching for similar driving conditions in the history) and in the latter case the reference is taken from a similar building (typical for the retail stores chains).

Finally, towards designing intelligent BEMS, a variety of control design optimization approaches exist, facilitating a vast diversity in the shape of the controller, the necessity of a model of the building, the required inputs and so on. Despite the differences between these approaches, the general control problem can be described by defining some basic components (for a detailed description see Kontes et al. 2012b). To start, let's consider that the physical system (building) can be described by a thermal simulation model, which is able to predict the thermal state of the actual building, taking into account the current building states (like wall and air temperatures, humidity, etc.), the predicted weather conditions and the control actions (like heating and cooling loads, shading angle, etc.) applied to the building.

Having such a model at hand, allows designing series of control inputs that lead the physical building to a set of states that are (near-) optimal with respect to a performance measure, using the model for the design process. In buildings domain, the performance measure is modelled as a constraint optimization problem, facilitating the following components:

- A performance-indicating KPI, usually correlated with operational cost or energy consumption (e.g. minimization of the total energy consumption, minimization of the grid-supplied energy, maximization of the net energy produced, etc.).
- A set of KPIs acting as constraints, ensuring comfortable in-building conditions for the occupants (e.g. visual, acoustic, thermal comfort constraints, etc.).

The availability of the building model, along with the stochastic nature of the occupancy patterns and weather conditions consist the use of model-assisted control (or model-predictive control, (MPC)) design optimization techniques (Goodwin et al. 2005, Bertsekas 1995) suitable for solving the above constraint optimization problem – see (Ma et l. 2010, Oldewurtel et al. 2010, 2012, Giannakis et al. 2011, Pichler et al. 2011, Kontes et al. 2012a, Cigler et al. 2012) for successful application in buildings. Here, based on the available building model and weather predictions, the optimization problem is solved for the period of time accurate weather predictions are available (prediction horizon), while the resulting controller is applied for a shorter period of time, called the control horizon. Besides the predictions, this approach necessitates the availability of sensor data from the real building for the previous days, since they will be used for the *warming-up process*, i.e. the model will be simulated using the actual building conditions for the previous days, in order to assimilate the actual thermal state of the real building at the beginning of the optimization process.

3 BAAS SYSTEM

3.1 *Motivation*

The major obstacle for the deployment of new smart building control and monitoring technologies is the deployment cost. Typically, more advanced the technology is, more contextual information is required for the proper setup. In addition, since expert engineers are required for the task, the overall cost further increases. The reason is that due to lack of standardization (too many proprietary standards) in the building automation area, a lot of work has to be done manually requiring deep knowledge about the particular system. This problem is most evident in the commercial buildings domain where the need for advanced technologies grows quickly. Typically an engineer responsible for designing control algorithms decides rather to deploy safe but robust solutions – usually a simple rule based controller; and such solutions can hardly attain the performance of advanced control algorithms like MPC.

In ideal case, the customer should be able to select a software package capable of providing desired functionality (building energy management, fault detection and diagnostics, control optimization, etc.), and the local installation or remote connection (in case of SaaS business model) is done automatically, so the tool is enabled to be used in a short time – and this is exactly the way the BaaS is aiming at. The target is to reduce required human interaction during deployment to some necessary minimum, ideally handled by a friendly GUI which would cut the deployment costs down. BaaS is thus pioneering the major enabler of advanced building technologies, like MPC that were mainly used only in the industrial domain and unlike commercial solutions uses open standards to achieve it.

3.2 *Building Information Models*

Within BaaS project, the path selected for treating the aforementioned limitations is the use of properly populated Building Information Model (BIM) with all the contextual information about the building in a standard way, along with the development of tools enabling utilization of such information. Under this perspective, the use of Industry Foundation Classes (IFC) data model is used as a standard way of describing building contextual information.

The use of BIM and IFC tools allows for semi-automatic deployment and operation of APO services in all buildings at hand, regardless of variations on the building types, construction, location and available systems. On the other hand, the use of BIM and IFC alone inserts more complexity to the problem, rather than simplifying the task, since requiring by all software components to provide support for the entire IFC schema is not a viable solution (Bazjanac 2007). Due to this fact, the concept of Model View Definition (MVD) has been adopted within BaaS. An MVD defines the smallest possible subset of the full IFC schema required to satisfy one or many exchange requirements, thus, the exchange requirements for each APO service are defined and made publicly available.

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. So the “sending” component (let's call it the writer), should create all the information to be sent (in conformance to the MVD), and the “receiving” component (let's call it the reader), should know how to use the information (which comes in conformance to the MVD), to perform some useful task. So *both* the “reader” and the “writer” should be designed to satisfy the requirements posed by the MVD (i.e. understand the MVD).

Now, it is conceivable that there are many “writer” components, like CAD tools or GUI interfaces that populate aspects of the data model. BIM 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. we can expect some meaningful data in the response. This way a “library” of queries for each exchange requirement (FDD, CDO, etc.) can be generated, that will be automatically supported by all ifc files compliant to the MVD. Finally, following this approach, the APO service modules are equipped with auto-configuration capabilities, while new modules can be imported to the system through a trivial process, as long as they ensure compatibility with the exchange requirements of the respective service, i.e. they are MVD-compliant.

There are a number of software tools allowing querying and other manipulation of BIM data – the BaaS project focuses mainly the part of obtaining relevant data from BIM and using them efficiently in the building services aiming for control optimization, fault detection and diagnostics, etc. All such functionality should be enabled automatically, without any human interaction. BaaS can be seen as a platform enabling the building contextual data as well as dynamic data (sensor readings) to software modules providing the actual service or functionality needed. Such platform is supposed to enable required inputs to many other features and software modules and impact significantly the market, especially if it is open to public.

3.3 *Simulation models*

In addition to simulation as a design and decision-support tool, we take the stance within BaaS, that simulation is an essential ingredient to providing APO services. In this operational-phase utilization of simulation models, the usage scenarios are different: simulation models are consumed by APO services that provide useful functionalities with respect to the building operation. The availability of factual (sensed) data, along with forecasts for pertinent parameters (e.g. weather, occupancy) can be exploited and actively used to bridge the "simulated" and "real" worlds, reducing or even mitigating design-phase uncertainties.

With respect to the existing calculation methodologies for simulation models, quasi-static and CFD calculation methodologies are primarily useful in the design phases, either due to the resolution of their predictions (annual basis for quasi-static) or due to the inherent assumptions and modelling detail required – as such, they are of lesser importance within BaaS. The use of time-steps in the range of a minute to one hour allows to account for the dynamics of active climate control systems, but also to incorporate control strategies that use state measurements as inputs to compute actuation commands. The desire to use simulation as a forecasting tool, also suggests that a "small" time step might be warranted. In view of the comments above, in Figure 1, the type of calculation methodologies of interest to BaaS can be identified.

Simulations and their respective calculation methodologies will be used with BaaS to accomplish a variety of different tasks:

- **Energy performance estimation:** In this task the energy performance of the whole building is estimated. Energy performance includes total energy needs, including energy used for conditioning the spaces. In the transient calculation methodologies above thermal comfort parameters can also be computed.
- **Energy performance forecasting:** The goal of energy performance forecasting is to estimate building energy needs in order to preserve comfort conditions in building spaces, during a finite future time horizon. The use of forecast data obtained from various sources is necessary in this case. As it can be expected the validity of the forecasting process depends strongly on the quality of the forecasts. Integration of past data and forecast data (obtained from different sources) is essential here and the abstractions of the middleware will facilitate access to these data, so that the problem can be correctly set up.
- **Model calibration:** Although models are designed to predict the real behaviour of buildings and their systems as accurately as possible, their predictions may differ from real sensor measurements, due to a variety of reasons including: sensor measurement errors, modelling insufficiencies, or incorrect model parameter value's estimations. Model calibration tasks rely on past sensor measurements in order to change the model parameter values and bridge the above gap.
- **Components validation:** System performance can degrade over time, leading to out-of-specification operation. This can have adverse effects with respect to energy performance and thermal comfort. Anomaly detection and identification using simulation-based methodologies can be one of the ways, to identify such events
- **Control design:** The general purpose of (supervisory) control design is to design a controller that given state parameter values will return operation schedules and commands of controllable building elements. In model-based control design, the calculation methodology (here synonymous to "model") is used in combination with model-predictive control algorithms to generate such strategies.
- **Control design optimization:** The generated control actions, using simplified state-space models, can have poor performance when applied to the real system. For this reason, the resulting controllers can be improved using more "accurate" building models, by performing a second optimization step. Uses of calculation methodologies can be an invaluable asset in fine-tuning/optimizing controller parameters.

3.4 *System architecture*

Within the BaaS architectural design, a three layer architecture is envisaged: the data layer serving static and dynamic data needs through the implementation of an extended Building Information Model (eBIM) comprising a data warehouse and a BIM server; the communication layer acting as an abstraction layer to facilitate communication between the physical and ICT layers; and the APO Service Layer to provide the reasoning and analytics services. These functionally disjoint layers operate independently, communicating through the use of properly-defined (software) interfaces. The term APO collectively refers to continuously recurring tasks during building operation: assessment of the current building state; prediction of the effects that various decisions will have to KPIs; and, optimization of performance as measured through relevant KPIs.

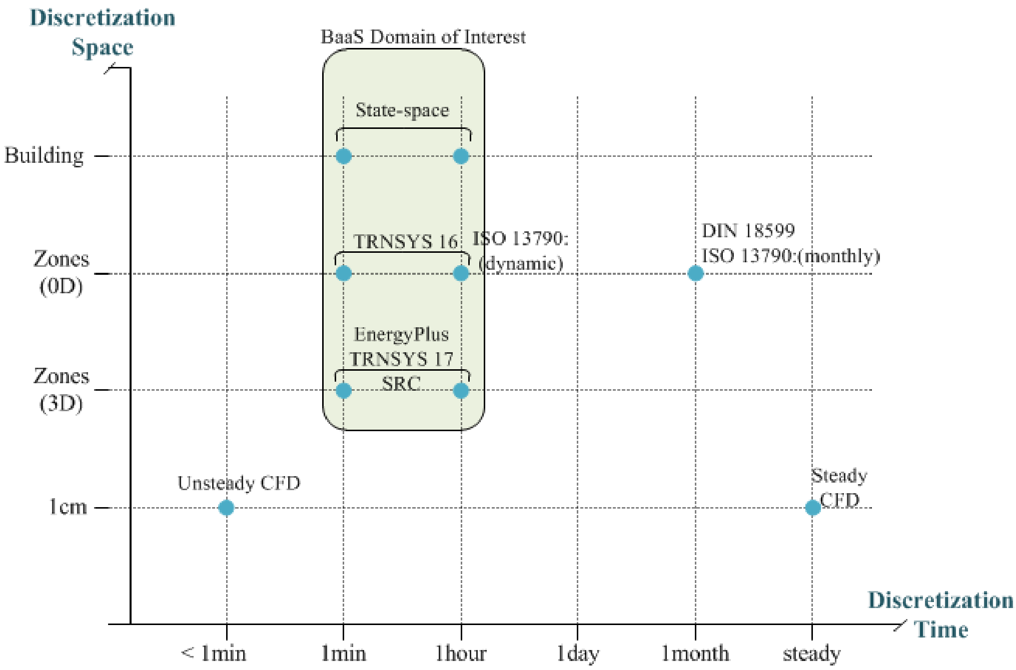

Figure 1. Calculation methodologies of simulation models.

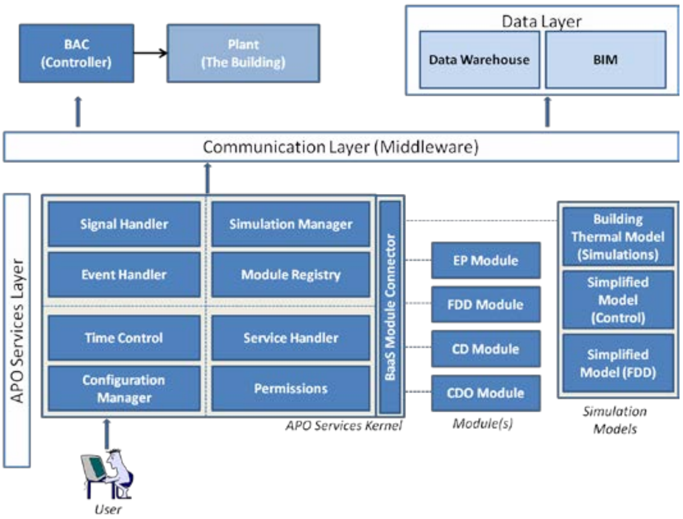

Figure 2. APO Services Layer – Core Components.

Functionally the APO services layer is intended to host the necessary algorithms to analyze the collected in-building data, interactions of processes, and generate control strategies for effective energy management. Specifically, the services will provide core intelligence, building/facility assessment and monitoring, prediction and optimization services, utilizing information made available by the data layer services. The aforementioned activities can be grouped together in three functionally-disjoint *groups of services*:

– **Fault Detection & Diagnostics**, containing services which provide analytics to detect and possibly find a root cause (diagnosis) of various equipment malfunctions and faults
– **Energy Management**, containing services which provide analytics to monitor equipment performance at various building hierarchy levels (from the building envelope to the individual building equipment) and to identify critical levels for effective operation in order to take measures for respective maintenance
– **Control Design and Optimization**, containing services which provide control-related analytics to design monitor and optimize applied control strategies by identifying control faults and inefficiencies.

At the APO Layer, what is collectively denoted as services should be understood as functional components implemented as a collection of software modules. These modules are either developed during BaaS project or can be provided by interested stakeholders to implement analytics (fault detection and diagnostics, control design, etc.). From a business perspective these modules can be part of the business intelligence and solutions portfolio provided as a service to building owners and occupants.

A schematic representation of the APO kernel, containing all the necessary functional components, is shown in Figure 2. The main modules of the APO kernel are:

– The **module registry**, where various modules are available for use by any APO service, since all APO services that will be deployed later on to the system will have to select and use control design, control design optimization, energy monitoring, fault detection and simulation modules that are available through the module registry library.
– The **simulation manager** is responsible for providing a fully functional simulation model of the building to any APO service that requests it.
– The **service handler**, which is responsible for coordinating all APO services, by invoking the control design, control design optimization and fault detection modules, through the control and fault detection managers.

- The **control manager**, which is responsible for any control-related action within BaaS.
- The **fault detection manger**, which is responsible for detecting and identifying as many problems as possible of the building actuating components. Based on the user inputs, or predefined default set of faults to be monitored, the fault manager instantiates and manages a variety of fault detection and identification (FDI) abstract components.
- The **signal handler**, which handles all data (past, present, future) coming through the middleware layer to the given APO services. A signal is a generic software abstraction used throughout the APO kernel able to accommodate any type of data.
- The **event handler**, which is responsible for the event management: distribution to selected blocks, priority handling, repeating mechanism for unacknowledged events, logging, event time-of-life, etc.
- The **BaaS connector**, which secures the data connection (data access layer) between the APO services kernel and modules.
- The **configuration manager**, which collects the user requirements (cost function formulation – KPIs, constraints) entered through simple GUI and forms the setup for each task solved by the APO services.
- The **permissions and user manager**, which takes care of the security aspects related to the APO services. Number of user profiles can be generated with different privilege levels. New users added to the system are then assigned by the selected user profile.
- The **time control**, which is responsible for the proper timing of all APO services kernel actions.

4 EXAMPLE

Consider the simple one-zone office building shown in Figure 3, located in Germany. The building, as shown in Figure 4, is equipped with a temperature, occupancy and a window contact sensor, while a sensor on the roof is used to measure the outside dry bulb temperature. Moreover, the building is served by a dedicated HVAC system. Finally, a BIM server containing the building description in IFC format is available, along with a Data Warehouse scheme that contains historical sensor measurements for the building.

During the summer period, a static rule-based controller is applied, facilitating the following rule: when there is a demand for cooling (room temperature above setpoint) during working hours of the building, the window should open if the outside temperature is at least 3°C lower than the room temperature, thus saving energy by shutting-down the HVAC system.

In an effort to capture behaviours that lead to energy leakage, the fault detection manager instantiates a fault detection/identification object pertinent to a fault description specified by user or experts during the BaaS configuration phase. In this setting, a fault is indicated if the related zone HVAC system is operating and the window is open at the same time, since this is an energy inefficiency that needs to be reported and treated. The fault detection object instantiates initially a relevant symptom object that will evaluate a specified set of rules to detect incorrect behaviour. If the symptoms supporting the fault hypothesis are observed, i.e. the HVAC is operating while the window is open, the fault likelihood is increasing and after a period of time (fault reasoning) it exceeds a predefined threshold and a fault event is generated by the fault detection object.

In order to automatically adapt the specific fault detection logic to any building type at hand, a set of queries to the available BIM server are required, with the following order:

1. Get all window contact sensors of the building.
2. For each contact sensor, identify the specific window it is mounted to.
3. For each window, get the room it belongs to.
4. Get the air terminal and air terminal box serving each room.
5. Get the upstream HVAC structure (AHU, chiller) for each terminal.

Figure 3. Architectural view of the building.

Once the querying process has been completed, a collection of fault detection objects as the one described earlier is created and associated with each contact sensor – HVAC pair, while, at the same time, is able to request historical operational data for the sensor measurements and HVAC operation from the DW. Note here that this fault detection process assumes that all involved entities work properly (window opening sensor, zone temperature sensor, cooling valves, etc.).

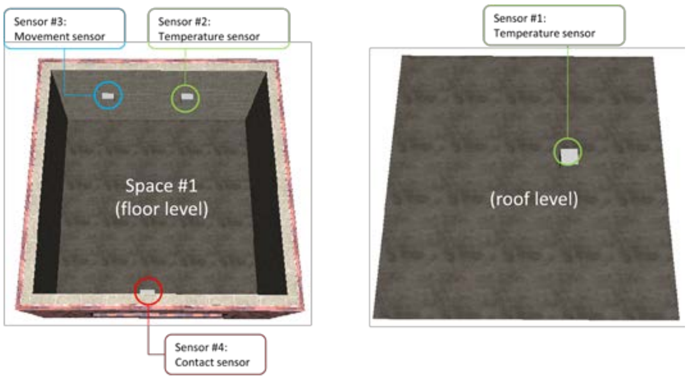

Figure 4. Available building sensors.

Since the entities dependencies are known through the querying process; the fault dependence object checks all relevant fault identification and detection objects outputs for possible faults. Should any involved entity be faulty, the FDI process must not be applied or its results must be discarded. An example of relevant objects tree is described in Figure 5, where the analyzed one zone example is put into wider context of the HVAC distribution system of a commercial building. The chiller (or chiller plant) is producing a cooling for the whole building, i.e. the chilled water is distributed to cooling coils in Air Handling Units (AHU) and optionally also in air terminal boxes serving the individual zones or rooms.

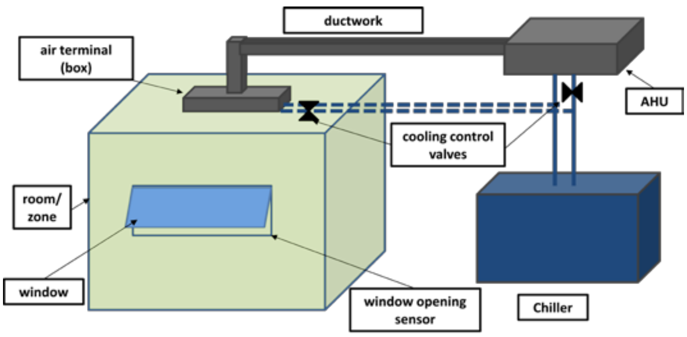

Figure 5. The HVAC system.

Assuming the fault detection object identifies a malfunctioning contact sensor; an event containing the information on the discovered fault is created and announced on the event handler and to the building manager. Subsequently, using pre-defined fault recovery logic (like a decision tree), the control manager postpones the use of the static rule-based controller and forces application of a model-assisted control design optimization module for the control of the zone. The new control approach assumes that the window is uncontrollable and attempts to regulate the HVAC operation in order to save energy and maintain comfort in the room.

For the new control design optimization problem, the warming-up period is set to two days, while the prediction horizon to three days. The new control function for the HVAC is a linear controller, transforming a set of building states (outside temperature, room temperature and occupancy) into setpoints, while the KPIs of the optimization problem are defined as the total energy consumption and the zone temperature.

In Figure 6, a draft sketch of the whole process is presented. First of all, the control manager requests historical data for the warming-up phase and reference occupancy data from the DW, as well as weather predictions from external services, through the middleware. The response signals are pre-processed and forwarded to the simulation manager. The simulation manager, using the simulation setup guidelines provided by the control manager, requests the available simulation model and creates the simulation and co-simulation objects, through which the historical, predicted and reference data are injected to the simulation, Finally, the control design optimization object created by the control manager is used to design the control strategy to be applied to the real building.

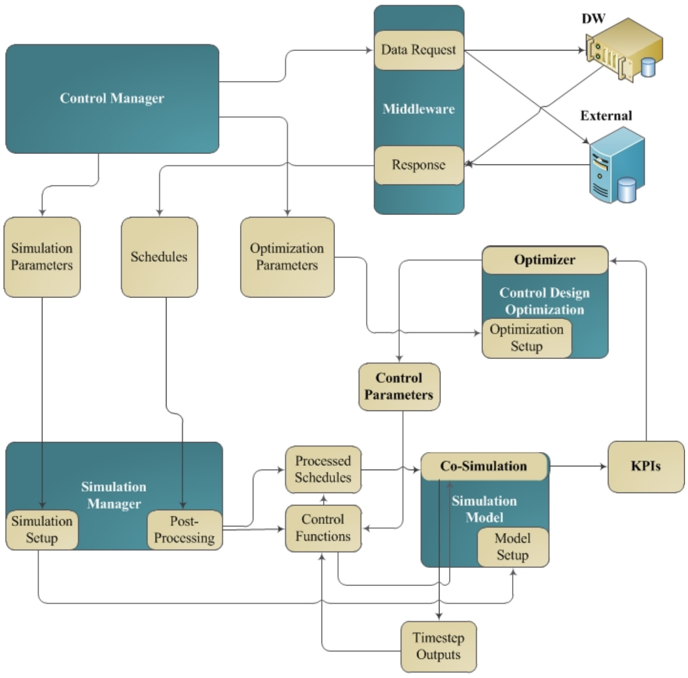

Figure 6. Functional flow chart for the one-zone simple example.

5 CONCLUSIONS

In the present work, the high-level architecture of the kernel hosting the BaaS APO services has been presented. BaaS system manages to consist a platform enabling the building contextual data as well as dynamic data (sensor readings) to software modules, allowing for semi-automatic deployment and operation of Asses, Predict and Optimize services in all buildings at hand, regardless of variations on the building types, construction, location and available systems, through the use of BIM and IFC tools and incorporation of MVDs.

6 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).

7 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*.

Bertsekas DP 1995. *Dynamic Programming and Optimal Control*. Athena Scientific

CAMPUS21 Project- Control and Automation Management of Buildings and Public Spaces, http://www.campus21-project.eu/index.php/en/

Cigler J., Prívara S., Váňa Z., Žáčeková E. and Ferkl L. 2012. Optimization of predicted mean vote index within model predictive control framework: Computationally tractable solution. *Energy and Buildings, 52, pp. 39-49,2012*

Du R., and Liang J. 2007. Model-based fault detection and diagnosis of hvac systems using support vector machine method. *International Journal of Refrigeration, Volume 30, Issue 6, September 2007, Pages 1104-1114*.

Giannakis G. I., Kontes G.D. , Kosmatopoulos E.B. and Rovas D.V. 2011. A model-assisted adaptive controller fine-tunning methodology for e_cient energy use in buildings. *In: 2011 Mediterranean Conference on Control & Automation*, pp 49 – 54.

Goodwin G.C., Seron M. and de Dona J. 2005. *Constrained control and estimation: an optimisation approach.* Springer Verlag, 2005.

HESMOS Project - ICT Platform for Holistic Energy Efficiency Simulation and Lifecycle Management of Public Use Facilities, http://hesmos.eu/

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

Kontes G.D., Giannakis G.I., Kosmatopoulos E.B. and Rovas D.V. 2012b. Demand-shifting using model-assisted control. *ICHMT DIGITAL LIBRARY ONLINE*, 2012.

Kukal J., Macek K., Rojicek J. and Trojanova J. 2009: From Symptoms to Faults: Temporal Reasoning Methods. *In the Proc. 2009 Int. Conference on Adaptive and Intelligent Systems*, Klagenfurt, Austria, 2009.

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).* 2010. pp. 5106-5011.

MOST Project – Monitoring System Toolkit, http://most.bpi.tuwien.ac.at/

PEBBLE Project - Positive Energy Buildings thru Better Control Decisions, http://www.pebble-fp7.eu/

Oldewurtel F., Parisio A., Jones C.N., Morari M., Gyalistras D., Gwerder M., Stauch V., Lehmann B. and Wirth K. 2010. Energy efficient building climate control using Stochastic Model Predictive Control and weather predictions. *Proceedings of the 2010 American Control Conference (ACC).* 2010, pp. 5100-5105

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*, 45:15-27, 2012.

OptiControl Project - Use of weather and occupancy forecasts for optimal building climate control, http://www.opticontrol.ethz.ch/

Pichler M.F., Dröscher A., Schranzhofer H., Kontes G.D., Giannakis G.I., Kosmatopoulos E.B. and Rovas D.V. 2011. Simulation-assisted building energy performance improvement using sensible control decisions. *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings,* pp. 61-66, ACM

Schein J., Bushby S.T., Castro N.S. and House J.M 2006. A rule-based fault detection method for air handling units. *Energy and Buildings Volume 38, Issue 12, December 2006, Pages 1485-1492*.