Schneider, G. F., Kalantari, Y., Kontes, G. D., Grün, G., and Steiger, S. (2017). “A platform for automated technical building management services using ontology.” In: *LC3 2017: Volume I – Proceedings of the Joint Conference on Computing in Construction (JC3)*, July 4-7, 2017, Heraklion, Greece, pp. 321-328. DOI: https://doi.org/10.24928/JC3-2017/0279.

A PLATFORM FOR AUTOMATED TECHNICAL BUILDING MANAGEMENT SERVICES USING ONTOLOGY

Georg F. Schneider[1], Yashar Kalantari[2], Georgios D. Kontes[3], Gunnar Grün[4], Simone Steiger[5]

Abstract: The deployment of technical building management services is a requirement to further reduce energy demand of future and existing buildings. Automating the process of configuring and deploying technical building management services such as fault detection and diagnosis of technical equipment seems to be a promising path to intensify the adoption of these services. In this work we present a data processing and analytics execution platform which allows the deployment of ontology-based, automated technical building management services on a large-scale. We present the platform architecture and results from a reference implementation performing rule-based fault detection on offline air handling unit data.

Keywords: Technical Building Management, Service-oriented Architecture SOA, Fault Detection and Diagnosis, Ontology.

1 Introduction

The buildings sector is a determining factor when addressing energy efficiency goals set by authorities (EuPC 2009). In particular, in non-domestic buildings, such as offices or hospitals, systems to provide acceptable thermal comfort conditions consume 30-40% of the final energy (Pérez-Lombard et al 2008). Up to 20% of this energy is lost due to poor control configuration and system faults (Roth et al 2005).

In the life-span of buildings such faults naturally occur as technical installations in buildings tend to deteriorate over time, e.g. from component failure or degradation. To address this problem, Technical Building Management (TBM) services such as Fault Detection and Diagnosis (FDD) are deployed and stipulated by standards (ISO 16484 2010; EN 15232 2013), which include all services and activities to ensure seamless operation of technical building systems. FDD services use monitoring data acquired through a Building Management System (BMS) to detect faulty behaviour of technical equipment, determine its root cause and estimate its impact on energy demand and/or costs (Venkatasubramanian et al 2003; Katipamula and Brambley 2005).

For the deployment of TBM services on large-scale, e.g. district level, ICT solutions offering scalability are required. Such solutions (software platforms) are usually cloud-

[1] Research Associate, Fraunhofer IBP, Nürnberg, Germany, georg.schneider@ibp.fraunhofer.de

[2] Software Architect, Fraunhofer IBP, Nürnberg, Germany, yashar.kalantari@ibp.fraunhofer.de

[3] Research Associate, Technische Hochschule Nürnberg and Fraunhofer IBP, Nürnberg, Germany, georgios.kontes@ibp.fraunhofer.de

[4] Professor, Technische Hochschule Nürnberg and Vice-Institute Director, Fraunhofer IBP, Nürnberg, Germany, gunnar.gruen@ibp.fraunhofer.de

[5] Group Leader, Fraunhofer IBP, Nürnberg, Germany, simone.steiger@ibp.fraunhofer.de

based in order to be able to cope with the amount of data, as well as facilitate the transparent deployment, management and execution of services.

Platform solutions for data and information management on a large scale have been proposed in the past. For example the goal of the IntUBE system (Böhms et al 2010) is to improve the energy efficiency of buildings and neighbourhoods by integrating information from Building Information Modelling (BIM), simulation and BMS. A Monitoring System Tool (MOST) is presented by Zach et al (2012) which allows to acquire monitoring data from different BMS systems and analyse and visualise it. Valmaseda et al (2013) report an event-driven Service-oriented Architecture (SOA) platform which features the integration of BIM and BMS data and the deployment of services for decision support. In particular for the case of Air Handling Units (AHU), a cloud-based solution for automated FDD through manual gathering of BIM information is presented by Bruton et al (2014).

Apart from open-source or project-based solutions, several vendors from industry offer integrated solutions for FDD services in buildings, with the majority being closed, proprietary systems, e.g. ATTUNE® from Honeywell, PANOPTIX from Johnson Controls Technology Company, and SKYSPARK from SkyFoundry, LLC (Dibowski et al 2016).

Even though significant effort has been made for defining complex TBM services and advanced hosting platforms, the problem of initial configuration and deployment of the analytics has been rather overlooked. Here, an initial mapping between the building static and dynamic data and the analytic inputs has to be performed manually – which is a time-consuming and error-prone process. In this direction recent approaches aim at automating this process by leveraging on ontology-based formal descriptions to integrate BIM and BMS information (Schneider et al 2016; Dibowski et al 2016). To enable a wide-spread adoption of energy-related TBM services, ontology-based, automated approaches seem to be a promising technology as the required information is published more and more using standardised data formats, e.g. from BIM methodology.

The contribution of this work is the architecture and prototype implementation of a holistic solution which: (1) supports the integration of the aforementioned ontology-based TBM services following the contemporary SOA paradigm; and (2) handles the deployment, management and execution of these services in a transparent manner. To ensure vendor independence, the proposed solution builds on open-source software and frameworks for the implementation.

After describing the general architecture of the platform in the next section, we present a reference implementation of the platform and discuss results obtained for the use case of rule-based fault detection for offline AHU data.

2 Platform Architecture

2.1 General Architecture

Here, the Application Integration Platform defined in MOEEBIUS H2020 European project (MOEEBIUS Project) has been utilised (Kontes et al 2016). MOEEBIUS introduces a Holistic Energy Performance Optimisation Framework that enhances current modelling approaches and delivers innovative simulation tools in an effort to reduce the gap between predicted and actual energy consumption in buildings.

The overall software architecture (Kontes et al 2016) is illustrated in Figure 1. The main design principles considered are:

- Service-oriented Architecture: Any functionality realised is implemented as a micro-service, which communicate via messages; This supports reuse and recombination to complex processes within the Enterprise Service Bus (ESB);

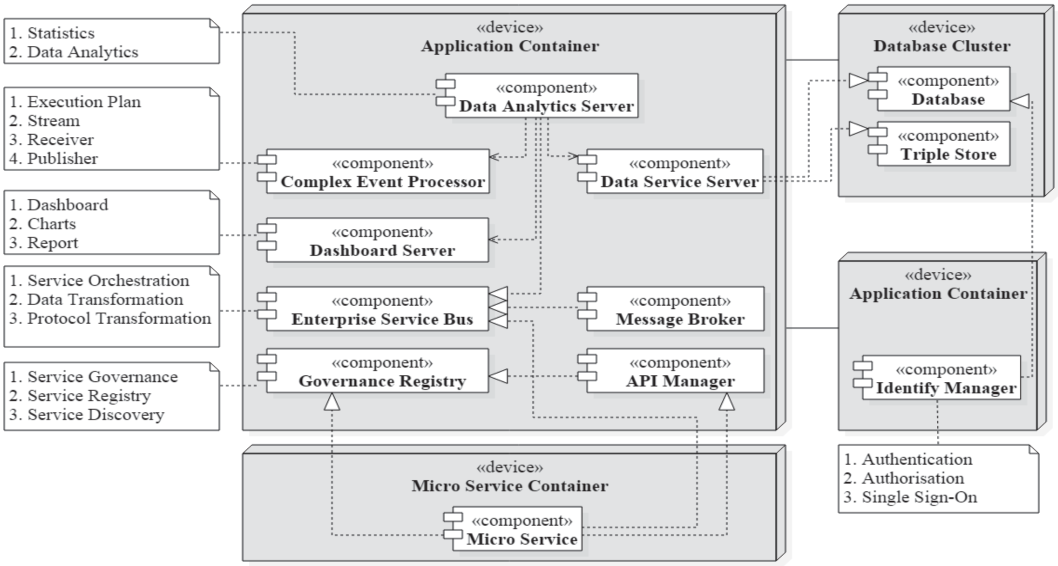

Figure 1: Software architecture of platform adapted from Kontes et al (2016).

- Data communication paradigm supporting two ways of communication: (1) communication between micro-services through ESB, which enables easy monitoring and controlling of interaction among services; (2) event-based communication through certain endpoints, needed for processing of data streams;
- Data storage is scalable depended on the needs of the applications; the platform supports relational and non-relational databases; As an extension of Kontes et al (2016) a triple store is integrated to store static BIM and BMS data by means of ontology;
- Analytics processing the available data can execute as batch process in the ESB or as near real time analytics on stream data;
- Visualisation of reports and dashboards in form of plots and charts;
- Ensuring a secure access to the platform and its functionalities through a unified security solution through authentication, authorisation and encryption.

The definition of these design principles is based on the distillation of the functional and non-functional requirements posed within MOEEBIUS project and are described in detail in Romero et al (2016).

For implementing the designed architecture, within MOEEBIUS project it has been decided to adopt the SOA middleware WSO2 (http://wso2.com) rather than implementing a custom solution from scratch. WSO2 covers the designed functional and non-functional requirements and is available open source. Many components are available to be used 'off-the-shelf' or to be adapted. The solution can be deployed in private cloud on premise or public cloud.

2.2 Architectural Design of Ontology-Based FDD Micro-Service

Formalising domain information using ontology allows for automatic configuration and deployment of TBM services, such as rule-based FDD (Schneider et al 2016). The methodology to integrate the ontology-based TBM services in the software platform described above is depicted in Figure 2, using the APAR rule set (Schein 2006) as an example. The detailed ontology and its mappings to BIM models is out of the scope of this work and presented in detail in Schneider et al (2016).

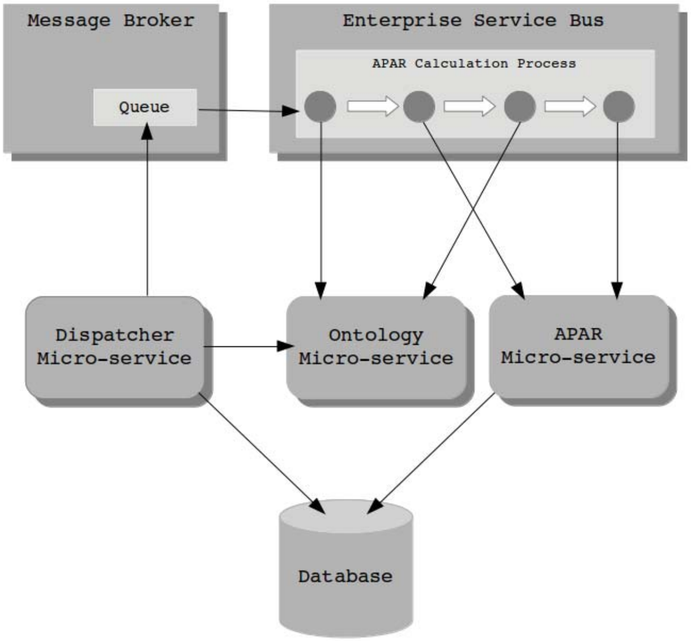

Figure 2: Architecture for platform integration of ontology-based TBM services.

All functionalities, i.e. triple-store, TBM service and dispatcher (Figure 2: Ontology Micro-service and APAR Micro-service and Dispatcher Micro-service, respectively) are published as micro-services and integrated overall in the ESB. The dispatcher triggers the execution of the services and can be customised to the requirements of the TBM system (e.g. time or event-based execution). When calling the dispatcher, a message is sent to the Message Broker which stores messages containing execution jobs and runs each TBM service depending on current free resources on the deployment platform. The actual execution of the TBM services is defined as a sequence of service calls in the ESB of the platform. From there, instances of the described micro-services are called which communicate with a database for retrieving and storing data. Using a message broker to handle queuing and execution of jobs offers the benefit of allowing distributed and parallel execution. This improves the performance of the system significantly when processing a large amount of data or using multiple TBM services.

3 Implementation

For demonstration purposes, we present results from performing rule-based fault detection on offline AHU data. We implement the APAR rule set (Schein 2006) in Java and integrate this code along with its automated, ontology-based configuration and deployment (Schneider et al 2016) as a micro-service in the platform using the previously defined architecture.

The APAR rule set consists of 28 rules which can be used for fault detection in AHU operation when the AHU is in steady state. Each rule is formed by a set of mathematical expressions evaluated for each time step; if a rule is evaluated to true this is considered as a symptom of a fault (Trojanová et al 2009). The number of faults is integrated over time to calculate a probability of fault for a certain time interval. An activity diagram detailing the process is depicted in Figure 3.

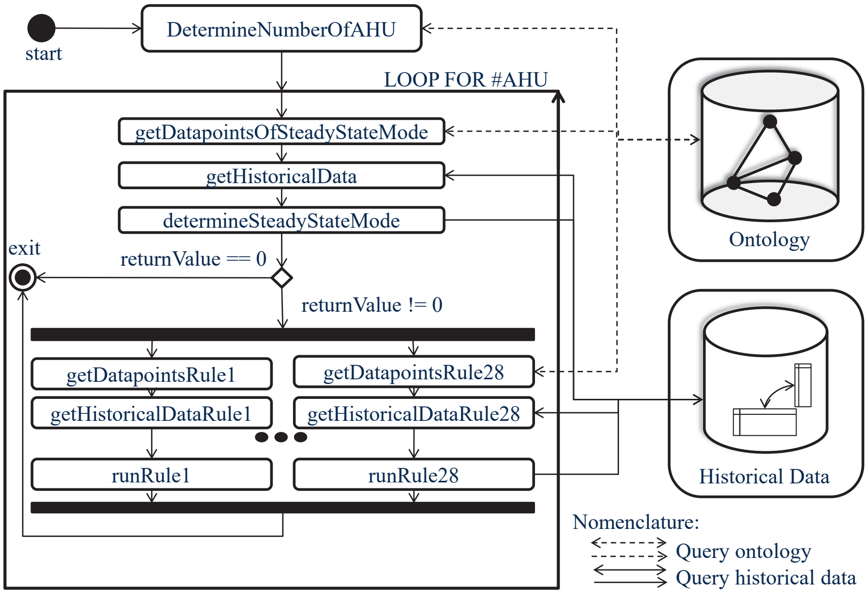

Figure 3: Activity diagram of executing the APAR rule set micro-service.

Here, the following sequence of execution takes place: (1) retrieve required identifiers from the ontology through a customised query; (2) retrieve and process historical data; and (3) store calculated results. Initially, the total number of AHUs is determined from a custom query to the ontology. Then the steady-state operation of the current AHU is determined, i.e. the control mode is constant for one hour in either (Schein 2006): heating; cooling with outdoor air; mechanical cooling with 100% outdoor air; mechanical cooling with minimum outdoor air; or unknown mode. If the AHU under analysis is in steady state the corresponding rules are executed.

For evaluation of the system we use measurement data from three AHUs installed in Richland, USA freely available online (DOE 2015). The data set holds values for temperatures and normalised control signals in equidistant, one-minute time intervals. These are exactly the inputs required by the APAR rule set. The data set contains measurements from July 2014 till June 2015.

We process the data with the solution as described above. The rules have been parameterised with default values as suggested by Schein (2006). The results include for each AHU the specific rule which is violated, the time stamp of the interval where the probability of a fault is greater than zero, the current operation mode and the value of the probability of a fault. For example, exactly 1000 hourly intervals in the data set of AHU 1 have a probability of fault greater than zero with faults detected in heating, cooling with outdoor air, mechanical cooling with 100% outdoor air or unknown mode.

Detecting a fault is one task to solve; identifying the root cause of a fault is a different and more complex task. In Figure 4**Error! Reference source not found.** the measured control signals and temperatures of AHU 1 for a faulty, detected time interval are plotted.

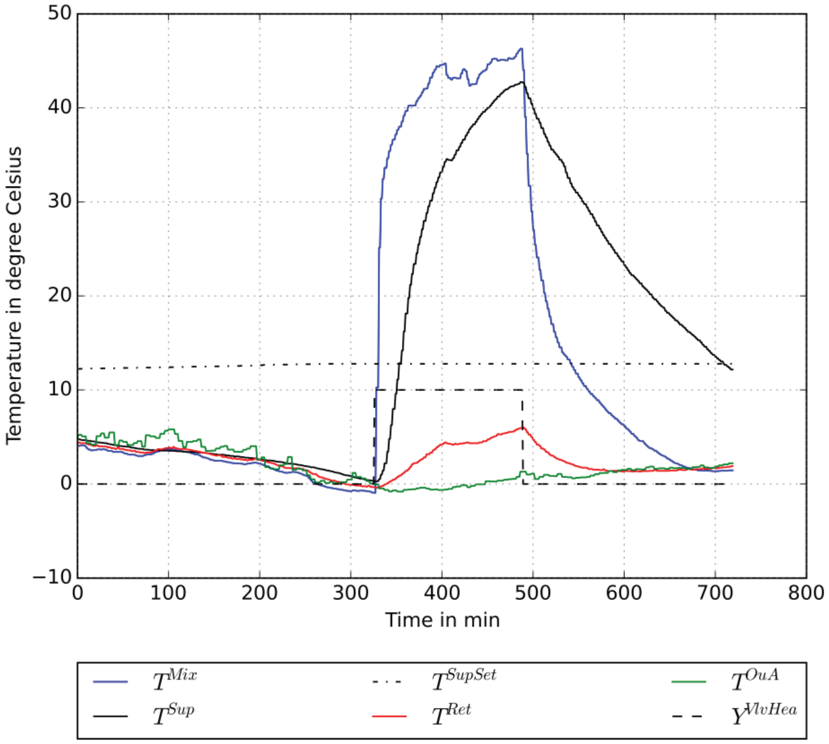

Figure 4: Temperatures and heating valve signal for a faulty interval 2014-12-07 start at 04am (middle of plot at 400min); $\mathrm{T}^{\mathrm{Mix}}$: mixed air temperature, $\mathrm{T}^{\mathrm{SupSet}}$: supply air temperature setpoint, $\mathrm{T}^{\mathrm{OuA}}$: outdoor air temperature, $\mathrm{T}^{\mathrm{Sup}}$: supply air temperature, $\mathrm{T}^{\mathrm{Ret}}$: return air temperature, $\mathrm{Y}^{\mathrm{VlvHea}}$: Normalised heating valve control signal.

In the middle of the plot (400 min) the possibly faulty interval starts. The system detects a violation of APAR rules 1, 2 and 4 with the system in heating mode. . Within the regarded time frame the heating coil valve opens fully for about 120 minutes leading to a maximum value of the supply air and mixed air temperature of about 43 degree Celsius.

The APAR suggests in this case as possible interpretations sensor errors, stuck valves, stuck dampers or problems with the circulating pump. From our experience a different, possibly valid interpretation applies. Usually in AHU control if the outdoor air temperature reaches zero degrees Celsius, the control logic heats up the heating and cooling coils to prevent damage from freezing. For this certain case with the limited information available identifying the real root cause of the detected fault remains difficult.

4 Discussion

For the described use case the solution yields plausible results. However, some limitations exist.

The number of FDD methods deployed on the platform is rather limited. We intend to expand the scope of the platform in terms of the variety of FDD analytics for different technical systems, as well as enriching it with more types of analytics, targeted for control optimisation, energy prediction and benchmarking, etc. In addition, the proposed solution will be evaluated with monitoring data from other facilities and technical equipment.

Since the underlying formal semantics of ontologies allow for intelligent applications enabled through reasoning, future research should focus on this area of interest. Examples are isolating the root cause of a fault (Dibowski et al 2016) or supporting proactive and predictive maintenance of technical equipment in a building.

5 Conclusion

Within this paper we present the architecture and a reference implementation of a platform for the automated deployment of ontology-based Technical Building Management (TBM) services. Based on the defined architecture any functionality is implemented as a micro-service (Service-oriented Architecture - SOA) allowing the transparent execution, reuse and recombination to complex processes. We present results for deploying rule-based fault detection to offline AHU data for a full year implementing the APAR rule set (Schein 2006) as a micro-service.

In future we intend to expand the capabilities of the solution in terms of TBM services supported by adding more FDD methods applicable, e.g. for boilers. As the solution is based on open-source software and frameworks it may be integrated with existing Internet of Things/Big Data platforms. Furthermore, an extensive testing and validation of the proposed solution is required also to integrate it within overall building management activities in a target facility.

6 Acknowledgments

This research was performed as part of the Energie Campus Nürnberg and supported by funding through the “Aufbruch Bayern (Bavaria on the move)” initiative of the state of Bavaria. Georgios Kontes and Gunnar Grün gratefully acknowledge financial support from MOEEBIUS project, a Horizon 2020 research and innovation program under grant agreement No. 680517.

7 REFERENCES

Böhms, H.M., Plokker, W., Charvier, B., Madrazo, L. and Sicilia, A. (2010). IntUBE Energy Information Integration Platform., ECPPM, Cork, Ireland.

Bruton, K., Coakley, D., Raftery, P., Cusack, D.O., Keane, M.M. and O’Sullivan, D.T.J. (2015). Comparative analysis of the AHU InFO fault detection and diagnostic expert tool for AHUs with APAR. *Energy Efficiency*, 8(2), pp. 299–322.

EN 15232 (2013). *Energy performance of buildings – Impact of Building Automation, Controls and Building Management*, CEN, Brussels, Belgium.

Dibowski, H., Vass, J., Holub, O. and Rojíček, J. (2016). Automatic setup of fault detection algorithms in building and home automation., ETFA, Berlin, Germany, pp. 1–6.

DOE (2015). *Long-term data on 3 office air-handling units*. [Accessed 01 Jan 2017]. Available at https://trynthink.github.io/buildingsdatasets/.

EuPC (2009). Decision No 406/2009/EC. European Commission and European Parliament. http://data.europa.eu/eli/dec/2009/406/oj (permanent link).

International Organization for Standardization (ISO) (2010). *ISO 16484: Building automation and control systems (BACS)*, Geneva, Switzerland.

Katipamula, S. and Brambley, M.R. (2005). Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—A Review, Part I. *HVAC&R Research*, 11(1), pp. 3–25.

Kontes, G.D., Kalantari, Y., Tsatsakis, K., Papapolyzos, D., Papapolyzos, T., Lazaropoulou, M., Tsitsanis, A., O'Flynn, B., Aherne, S., Nales, A., van Tooren, P., Beder, C., Klepal, M., Malavazos, C., Agustin, P. de, Armijo, A., Tellado, B. and Sanchez, V. (2016). D3.1 MOEEBIUS Framework Architecture including functional, technical and communication specifications. Project Deliverable, MOEEBIUS project.

Pérez-Lombard, L., Ortiz, J. and Pout, C. (2008). A review on buildings energy consumption information. *Energy and Buildings*, 40(3), pp. 394–398.

Romero, A., Tellado, B. and Tsitsanis, T. (2016). MOEEBIUS energy performance optimization framework in buildings for urban sustainability. IAHS, Albufeira, Portugal, pp. 1–10.

Roth, K.W., Westphalen, D., Feng, M.Y., Llana, P. and Quartararo, L. (2005). *Energy impact of commercial building controls and performance diagnostics: market characterization, energy impact of building faults and energy savings potential*. Technical Report, TIAX LLC.

Schein, J. (2006). Results from field testing of embedded air handling unit and variable air volume box fault detection tools (NISTIR 7365). US Department of Commerce, Technology Administration, National Institute of Standards and Technology, Washington, D.C., USA.

Schneider, G.F., Kalantari, Y., Kontes, G.D., Steiger, S. and Rovas, D.V. (2016). An Ontology-Based Tool for Automated Configuration for Automated Configuration and Deployment of Technical Building Management Services., CESBP/BauSIM, Dresden, Germany, pp. 865–872.

Trojanová, J., Vass, J., Macek, K., Rojiček, J. and Stluka, P. (2009). Fault Diagnosis of Air Handling Units. IFAC, Barcelona, Spain, 42(8), pp. 366–371.

Valmaseda, C., Garcia, M.A., Hernandez, J.-L., Katsigarakis, K.I., Kontes, G.D. and Rovas, D.V. (2013). An event-driven soa-based platform for energy-efficiency applications in buildings. CIB W78, Beijing, China.

Venkatasubramanian, V., Rengaswamy, R., Yin, K. and Kavuri, S.N. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3), pp. 293–311.

Zach, R., Glawischnig, S., Hönisch, M., Appel, R. and Mahdavi, A. (2012). MOST: An open-source, vendor and technology independent toolkit for building monitoring, data preprocessing, and visualization., ECPPM, Reykjavik, Iceland, pp. 97–103.