AN ONTOLOGY-BASED TOOL FOR AUTOMATED CONFIGURATION AND DEPLOYMENT OF TECHNICAL BUILDING MANAGEMENT SERVICES

G. F. Schneider[1,2], Y. Kalantari[1,2], G. Kontes[1,2], S. Steiger[1,2], and D. Rovas[3]
[1]Fraunhofer Institute for Building Physics IBP, Nürnberg, Germany
[2]Energie Campus Nürnberg, Nürnberg, Germany
[3]University College London, London, United Kingdom

ABSTRACT

In this work we present the architecture and a reference implementation of a software tool to automate the configuration and deployment of services for fault detection and diagnosis which can assist to improve operational building performance. We use an ontology as an intermediate meta-data layer to integrate the BIM information and the static BMS data required to automate this process. The contribution aspect resides in the demonstration of the approach which is based on the core concept of a tailored pairing of a fault detection and diagnosis service and an ontology query specifically designed by experts. Suggestions for the architecture are provided on how to embed the ontology in the overall framework.

We present results from a first prototype to detect faults in real trend data of an air handling unit using rule-based analytics.

INTRODUCTION

The building sector accounts for the largest share of primary energy demand in industrialised countries, e.g. 43% of the German primary energy demand (AGEB 2014). With demands of about 30-40% of the required final energy in non-domestic buildings (Pérez-Lombard et al. 2008), HVAC systems have a significant impact on the overall energy efficiency of a building. Due to degradation of technical equipment, inadequate maintenance practices or poor operation schedules, the operational performance of buildings tends to deteriorate over lifetime. This is identified as one of the main causes for deviations observed between predicted and actual performance of a building (de Wilde 2014).

Technical Building Management (TBM) services such as Fault Detection and Diagnosis (FDD) use data gathered through a Building Management System (BMS) to enhance the operational performance of buildings. The use of these services is stipulated by related standards such as EN 15232 (EN 15232 2013). Various FDD methods with different complexity have been developed in the past (Venkatasubramanian et al. 2003a, 2003b, 2003c). Note, that in the rest of this paper we will use the term "analytic" for each instantiation of these methods.

Typically, manual input from experts which have analysed the system under control is necessary to configure and deploy such analytics. Bruton et al. (2014) present a comprehensive review on automating FDD for air handling units (AHUs) and recommend to improve the integration of static BMS data for future research activities, as often these are gathered manually by surveying the site.

In literature various meta-descriptions of Building Automation Systems (BAS) and facilities are reported. The definition of a common vocabulary, despite not formally specified, for BAS is provided by Domingues et al. (2016). Schein (2007) presents an information model for BAS implemented in the modelling language EXPRESS covering descriptions of devices, sequence of control functions and network topology of BAS. Ploennings et al. (2012) report an ontology for BAS termed "BASont" which is based on a concise description of BAS devices and their related semantics. Han et al. (2015) describe a rule-based reasoning system for detecting energy waste operation status of a ventilation system. The system is based on a description of the building energy system using an ontology and energy waste contexts may be inferred based on the model. A set of ontologies related to the "representation of energy-related information in future smart homes" is presented by Kofler et al. (2012). Corry et al. (2015) describe a "performance assessment ontology" which acts as a middleware to link information from various sources, such as building topology from Building Information Modelling (BIM), sensor information and building performance simulation. Tomašević et al. (2015) present an ontology for infrastructure facilities. The model is embedded in an energy management system applied in an airport test case.

The contribution of this work resides in providing a tool to automate the configuration and ease deployment of FDD analytics. Integral feature is the use of an ontology to integrate static meta-data needed to automate the process by integrating BIM information and static BMS data. The process involves the design of a tailored pairing of an analytic and ontology query based on the information requirements of each analytic. This approach is promising to circumvent error-prone and time consuming manual gathering of building static information as described in previous work

(Bruton et al. 2013). This is possible as information available in a standardised BIM format such as the Industry Foundation Classes (IFC) (buildingSMART 2015) can be linked to the ontology and retrieved automatically using built-in functionalities.

In the following sections first we briefly review methods for FDD in buildings as these define the information requirements of the designed ontology. The domain specific ontology for automated FDD is presented in the subsequent section. Next we describe the structure and components of a software tool designed to implement the prior mentioned method for automated configuration and deployment of TBM services. Finally we present insights from a pilot implementation and its deployment to fault detection in AHU operation.

A BRIEF REVIEW ON METHODS FOR FAULT DETECTION AND DIAGNOSIS IN BUILDINGS

FDD analytics (or services) involve the processes of detecting a fault (fault detection), determining the root cause of the fault (fault isolation) and evaluating the size of the fault and monitoring its impact over time (fault identification) (Katipamula and Brambley 2005).

The quality of a FDD system is determined by whether it possesses or not a set of desirable characteristics (Venkatasubramanian et al. 2003a), including, among others, quick detection of faults, isolability and adaptability. A plethora of methods have been developed in an effort to fulfil these characteristics, categorised in: (i) qualitative model based methods (Venkatasubramanian et al. 2003b); (ii) quantitative model based methods (Venkatasubramanian et al. 2003a); and (iii) process history based methods (Venkatasubramanian et al. 2003c).

In qualitative model based methods, a priori knowledge on the process governing the relationship between the inputs and the outputs of the system is described using qualitative causal models. In essence, they are expert systems facilitating a large set of if-then-else rules describing various faults.

Quantitative model based methods model the relationship between the inputs and outputs of the system using a mathematical formulation, like e.g. using a Kalman filter. Any deviation from the fault-free behaviour of the system, as predicted by the model, is marked as a potential fault.

In process history based methods, no expert knowledge is required; instead historical data describing the inputs and outputs of the system are utilised to construct data-driven models of the system, based on Pattern Recognition methods, such as Neural Networks. Again, any deviation from the predicted system behaviour is considered as a potential fault.

The intention of the methodology presented in this work is to enable the use of any type of the aforementioned methodologies in a transparent and laborious-free manner. To achieve this it is necessary to provide relevant meta-information for automated configuration, e.g. sensor type, location in building and affiliation to a technical building system.

A DOMAIN SPECIFIC ONTOLOGY FOR AUTOMATED FDD

Ontologies and Semantic Web Technologies

We use an ontology as a meta-layer to integrate the required information for the automated process in one abstract information entity. A frequently used definition of ontologies is: "An ontology is an explicit specification of a conceptualization." (Gruber 1993). In particular, ontological modelling defines a set of concepts and their relationships to describe a domain of interest. In comparison to other information modelling techniques, ontologies offer a unique advantage as they combine a high expressivity based on formalised abstractions with support of knowledge engineering methods arising from a firm formal base in Description Logic (Hitzler et al. 2010). In this work we use the term "ontology" to refer to an instance of a semantic data model.

Technologies and methods used for semantic modelling in the web are encompassed by the term "Semantic Web" which was introduced by Berners-Lee et al. (2001). The underlying data model utilised in this technologies is the Resource Description Framework (RDF) (W3C 2016a), where the smallest piece of information is modelled as a triplet consisting of subject, predicate and object, e.g. Sensor1 isAttachedTo Wall1. A modelling language which provides a vocabulary to formally specify a semantic data model is the Web Ontology Language (OWL) (W3C 2016b). A language to query RDF is SPARQL (W3C 2016c).

Domain Specific Ontology Description

The ontology presented here is developed following the approach suggested by Noy and McGuinness (2001). As a starting point, a critical review of existing ontologies was performed (Corry et al. 2015, Dibowski 2013, Domingues et al. 2016, Tomašević et al. 2015); four main contextual areas have been identified:

1) Parts and topology of a building and its systems;
2) Meta-data of technical building systems such as the technical affiliation to an automation entity, e.g. electric actuator to damper flap;
3) Physical aspects of BAS entities, e.g. physical sensor device or a gateway;
4) Virtual aspects of BAS entities, e.g. datapoint exposed through an application interface.

To enable exchangeability and to provide a consistent description of the building and its systems we integrate Meta Model an upper level ontology presented by Morbach et al. (2007). This allows the consistent description of objects and typing as well as part connectedness (i.e.: topology, e.g. beam 1 is connected to beam 2) and parthood (i.e.: mereology, e.g. room A is part of storey B) relationships among them. We use Meta Model as it provides exactly the needed upper level concepts while being light-weighted enough to be easily integrated. The integration of Meta Model results in a layered design of the developed ontology presented in Figure 1.

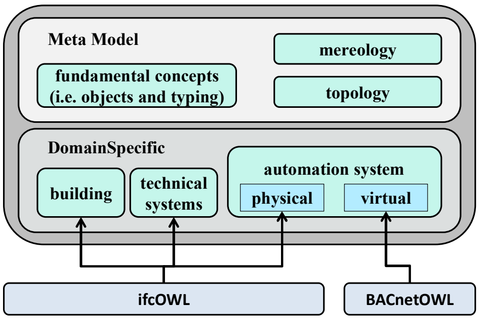

Figure 1: Layered structure of the developed ontology with upper level concepts from Meta Model (Morbach et al. 2007) and lower level concepts to describe building elements, technical systems and physical and virtual BAS abstraction. Black arrows indicate the linking of the ifcOWL and BACnetOWL ontology via concept mapping.

In the following, the taxonomy of the DomainSpecific ontology is described in more detail as depicted in Figure 2. In the ontology concepts are grouped in to *AutomationSystem*, *BuildingTopology* and *TechnicalBuildingSystem*. In general the paradigm of modelling separately concepts of an object and its type is applied. Information about a physical BAS component is abstracted using concepts summarised in *BasEntityPhysical*, e.g. a sensor is of type temperature. The concept *Virtual* groups on a technology independent level BAS data objects i.e. object abstraction and addressing. By this the concepts are provided to clearly differentiate between the physical virtual abstraction of a BAS as demanded by Domingues et al. (2016). Concepts for the topology of a building, e.g storey and room and of technical building systems such as fans, duct segments or boilers are specified using concepts from *BuildingTopology* and *TechnicalBuildingSystem*.

To integrate required BIM information and BMS data the DomainSpecific ontology is linked to a populated ifcOWL ontology (Pauwels and Terkaj 2016) and to a BACnetOWL, a self-developed ontology, via concept mapping (owl:sameAs) statements (Figure 1). A set of rules defined in the Semantic Web Rule Language (SWRL) is created to establish the necessary object and data property assertions by applying

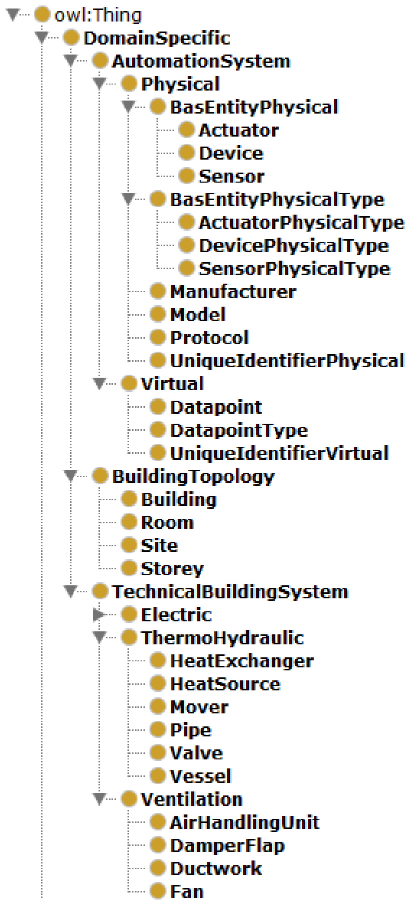

Figure 2: Taxonomy of DomainSpecific ontology.

a reasoner (W3C 2016d). The mappings established between different concepts of the utilized ontologies are listed in Table 1. As an example, we report in Table 2 the SWRL rule which defines the link of BIM and BMS data. The rule allows a reasoner to assert the object properties to link a physical BAS entity with its virtual datapoint counterpart and vice versa automatically by inference. This connection is established if literal values of the unique identifiers are equal (i.e. `swrlb:equal`).

Table 1: Mappings of ifcOWL (prefix: ifc) and BACnetOWL (prefix:bac) to the DomainSpecific ontology.

ifcOWL (ifc) of BACnetOWL (bac)	DomainSpecific Ontology
ifc:IfcSensor	Sensor
ifc:IfcActuator	Actuator
ifc:IfcSensorTypeEnum	SensorType
ifc:IfcActuatorTypeEnum	ActuatorType
ifc:IfcClassificationReference	UniqueIdentifierPhysical
ifc:IfcDuctSegment	Ductwork
ifc:IfcDamper	DamperFlap
ifc:IfcFan	Fan
ifc:ifcCoil	HeatExchanger
ifc:IfcValve	Valve
ifc:IfcUnitaryEquipment	AirHandlingUnit
ifc:IfcSite	Site
ifc:IfcBuildingStorey	Storey
ifc:IfcBuilding	Building
ifc:IfcSpace	Room
bac:Object	Datapoint
bac:UniqueIdentifier	UniqueIdentifierVirtual

Table 2: SWRL rule written in Protégé SWRLTabSyntax (Protégé CoP 2016) to link BMS and BIM information via the unique identifier value.

```
UniqueIdentifierPhysical(?UIp),
UniqueIdentifierVirtual(?UIv),
isIdentifiedBy(?BasPhys,?UIp),
isIdentifiedBy(?BasVirt,?UIv),
hasLiteralValue(?UIp,?IDp),
hasLiteralValue(?UIv,?IDv),
swrlb:equal(?IDp,?IDv) ->
isPhysically(?BasVirt,?BasPhys),
isVirtually(?BasPhys, ?BasVirt)
```

SOFTWARE TOOL FOR AUTOMATED CONFIGURATION AND DEPLOYMENT

In the following, first, we present the general workflow of a data integration and analytic execution tool with an application of the previously described ontology and we describe in detail and discuss functionalities of an analytics tool for configuring and deploying the respective analytics.

The general workflow of the tool is depicted in Figure 3. To initially setup the tool both static and dynamic raw data needs to be Extracted, Transformed to the final data format and Loaded into its respective storage (ETL).

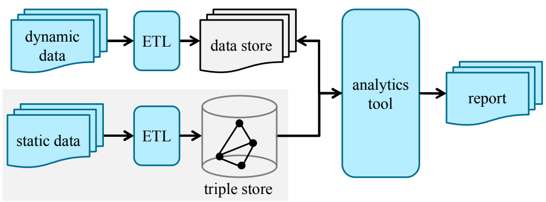

Figure 3: Structure of the software tool: Data Extraction, Transformation and Loading (ETL), storage, analytics configuration and deployment and reporting.

Static meta-information about the building structure and the installed technical building systems as well as their affiliated automation devices is extracted from raw data. The extraction of this so called BIM information may be done using arbitrary file formats or, if available, by standardised BIM data formats such as IFC. By providing a fixed mapping between concepts and SWRL rules this can be automated using reasoning. Similarly, static meta-information about the virtual objects of the automation system (e.g. the address of a virtual data object) is extracted from files specifying the configuration of the BMS and loaded to a connector ontology which is dependent on the communication protocol and technology utilised. For storage and providing access static data is loaded to a data base for its target format (i.e. RDF for an OWL ontology) termed *triple store*.

Dynamic data includes time trajectories of readings extracted from the BMS which can be in an arbitrary format depending on the BMS. This includes for example sensor readings or alarms. An ETL process extracts this data and loads it to a data store. The data store hosts the data and provides relevant data on request.

In both cases the ETL process includes steps for pre-processing the data, i.e. data cleansing for data quality enhancement, transformations such as unit conversions or changing from a native to a target format.

Performing the ETL process and setting up the static and dynamic data is a prerequisite for executing the analytics tool. The *analytics tool* actually performs the configuration and deployment of the analytics.

In the following we provide an explanation of the procedure to automatically configure and deploy an analytic by using an example of the configuration of a rule-based analytic with a value of a temperature sensor as input similar to the application in the test case described in the next section (see Figure 4).

- Step 1: Send a query to the triple store and process the acquired information about the sensor;
- Step 2: Join the static sensor information and its readings, i.e. extracting relevant time series data of sensor from dynamic data store;
- Step 3: Check if according to the rule a fault exists in each reading of the joined data.
- Step 4: Determine the probability of fault by dividing the number of occurred faults per each respective period of time;
- Step 5: Generate a result from fault detection

The query from the ontology and hence the retrieved information is specifically designed for each analytic. Also, the invocation of an analytic, e.g. schedule or event triggered, is a property of each analytic and needs to be specified along with the query and the job to execute the analytic. Multiple analytics result in a number of jobs each tailored according to the characteristics of the analytic. Nevertheless, the overall structure of the jobs remains similar comprising of querying the ontology, joining static and dynamic data, processing the analytic and reporting. If multiple instances of the same analytic are required, e.g. observation of temperature sensors in several rooms of a hotel, these are executed by passing parameters to the job templates and run each as a unique instance.

In case analytics require high performance either because of the amount of data processed or for performing computational intense calculations, e.g. data driven analytics such as machine-learning, parts of an analytic job may be delegated for distributed analysis to third party data analytic platform.

The results may be further processed for display in a suitable manner, e.g. energy dashboard or cost estimation tool. Technology-dependent information on the tool may be obtained from the pilot implementation documented in the last section of this work.

TEST CASE FOR AIR HANDLING UNIT OPERATION

To evaluate the functionality of the designed architecture and as a proof of principle a test case is designed on performing automated rule-based fault detection on offline AHU data. A comprehensive description of the general structure of AHUs as well as their modes of operation is provided by Schein et al. (2006).

Description of the Test Case

As *dynamic data* measurements are used of an AHU placed in Richland, USA available freely on the web (DOE 2015) gathered from July 2014 to June 2015. The data set contains readings for temperatures and set points stored with a time resolution of one minute. We select a period of nine days with outdoor air temperature below zero degrees Celsius in November 2014 and assume the AHU is in heating mode.

The mentioned dynamic data is shipped without additional information except a naming convention which allows human readers assign readings to datapoints. To demonstrate the automated extraction of *static data* from a BMS we use as a workaround BMS data of a single duct AHU installation on our premises. We do not expect an impact on the results generated reported here as the rules apply for generic single duct AHUs Schein et al. (2006). The installation utilises a BACnet automation system (ISO 16484-5 2014). We extract the information to populate the prior described ontology from parsing automatically generated spreadsheet files of the systems configuration. The BMS information, e.g. data objects and literal identifiers, are loaded directly to the BACnetOWL ontology. Required information is later integrated into the DomainSpecific ontology via the described methodology using SWRL rules and concept mapping.

We instantiate directly an ifcOWL ontology from parsing the spreadsheet files to extract the mereotopology of the system, i.e. BAS entity to technical device and building topology mapping. This step is necessary as current state-of-the-art BIM authoring tools have limited support to export automation system information unambiguously. For instance, the tool Revit 2016 (Autodesk 2016) exports both sensors and actuators using the IFC 4 Add. 1 data model (buildingSMART 2015) as *ifcBuildingElementProxy* instead of *ifcSensor* and *ifcActuator* causing ambiguity in the interpretation.

An example of the populated ontology after the data integration process is illustrated in Figure 5. We present the semantic relationships of the supply air temperature of the AHU. Via the object properties *hasTechnicalRelationshipTo* and *hasBuildingTopologyRelationshipTo* the individual representing the physical sensor is linked to the corresponding technical building system component *DuctSegmentSupply* and to the building topology element *Room16,* respectively. The connection to the virtual abstraction of the sensor covered by the individual *Object_124_AI* is established via the *isVirtually* object property. This relationship and its transitive companion *isPhysically* are established automatically by inference using the SWRL rule described in Table 2.

As an example *analytic* we present a fault detection test case, based on a simple rule for heating mode defined in Schein et al. (2006). Here, we only demonstrate the presence of an anomaly in the system, instead of performing symptomatic search to identify the root cause of faults, like e.g. in Trojanova et al. (2009).

For the purpose of this work we use the following rule (Schein et al. 2006).

$$\left|u_{heat} - 1\right| \leq \varepsilon_{heat} \wedge T_{sa,s} - T_{sa} \geq \varepsilon_{t} \quad (1)$$

The basic logic is that if the valve opening u_{heat} is nearly full opened in comparison to a threshold ε_{heat} and the difference between supply air temperature set point $T_{sa,s}$ and the supply air temperature T_{sa} is larger than threshold ε_t (usually ε_t = 1.2 K, Schein et al. (2006)) an anomaly is implied.

Following Trojanova et al. (2009), the rule can be interpreted as a symptom of a fault if the expression evaluates to true. Subsequently, we integrate over time to provide an estimate of the probability that there is a fault.

Results from Tool Execution on Real Data

Results from executing the described tool is presented in Figure 6 obtained by batch processing the selected time trajectories. We report the probability of fault for the mentioned time period of nine days. The probability of fault is calculated by evaluating eq. (1) for every minute of an hour and dividing the number of minutes where the expression is true by 60. We assume a fully open heating coil valve stipulating a fault when the supply air temperature deviates more than ε_t from its set point. In the analysed data set dissatisfying parameterization of the heating coil valve control causes overshooting of the supply air temperature, which is causes a high probability of fault. The end-result includes all faulty time intervals along with meta-information of sensors, e.g. location and identifier.

We implement the execution of the automated configuration of the described rule-based analytic using Talend Open Studio for Data Integration (Talend 2016). Pre-processing raw data of the time trajectories and static information in the sample are only some megabytes and have been undertaken using the statistical computing software environment R (R Core Team 2014). The populated ontology is stored in a Jena Fuseki server, i.e. triple store, which allows to query the ontology via a HTTP with SPARQL queries from Talend (Jena 2016). The execution of the tool in this test case requires on a standard desktop PC seconds.

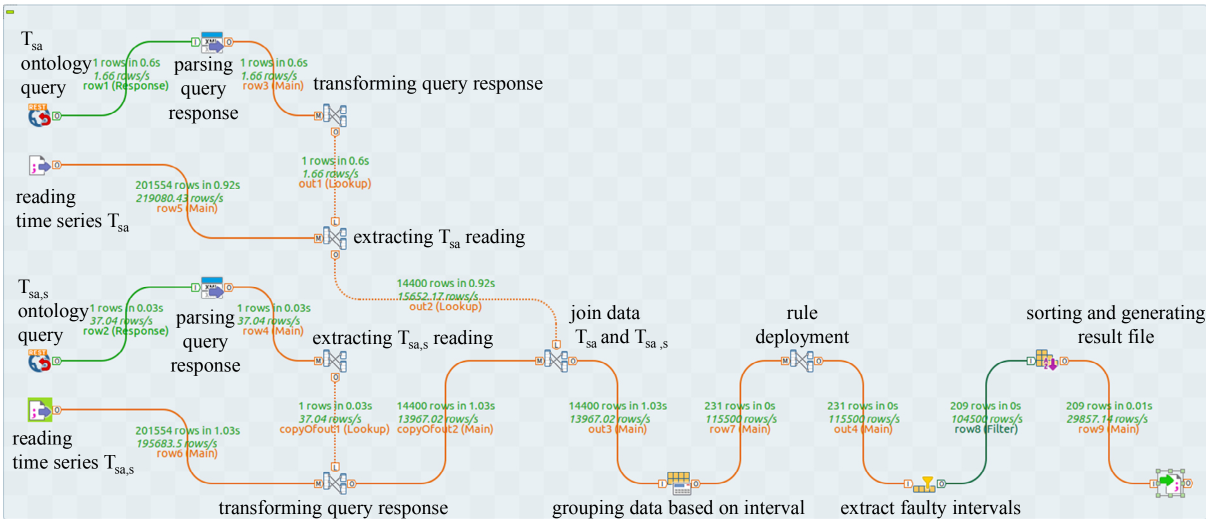

Figure 4: Screenshot of a job implementing the configuration and deployment of a rule-based analytic (see eq. 1) in Talend (Talend 2016). The job comprises querying for each sensor, joining static and dynamic data, deploying the rule-based analytic and creating reports. Processing speed of each step in row/s.

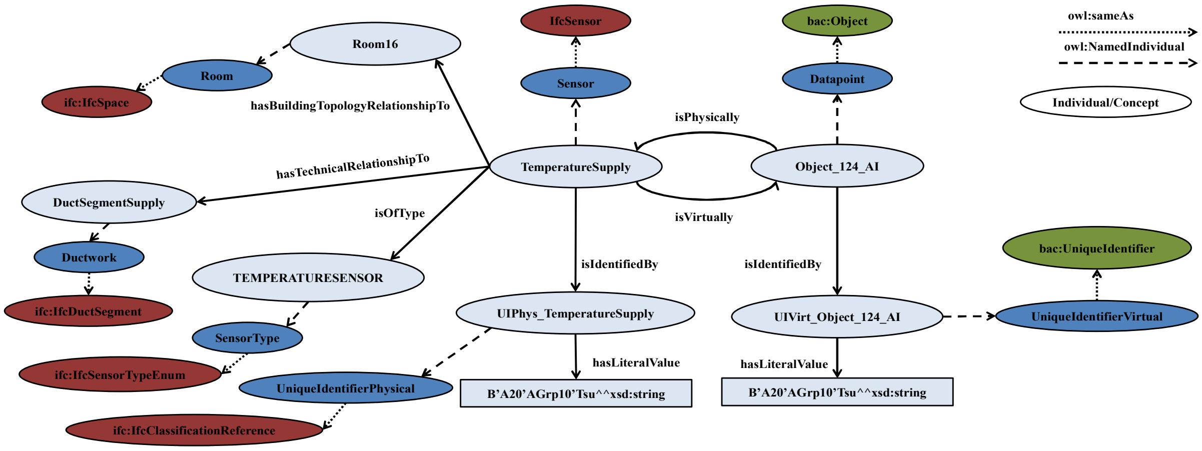

Figure 5: Semantic relationships of supply air temperature. Mappings of ifcOWL (prefix: ifc) and BACnetOWL (prefix: bac) to the DomainSpecific ontology.

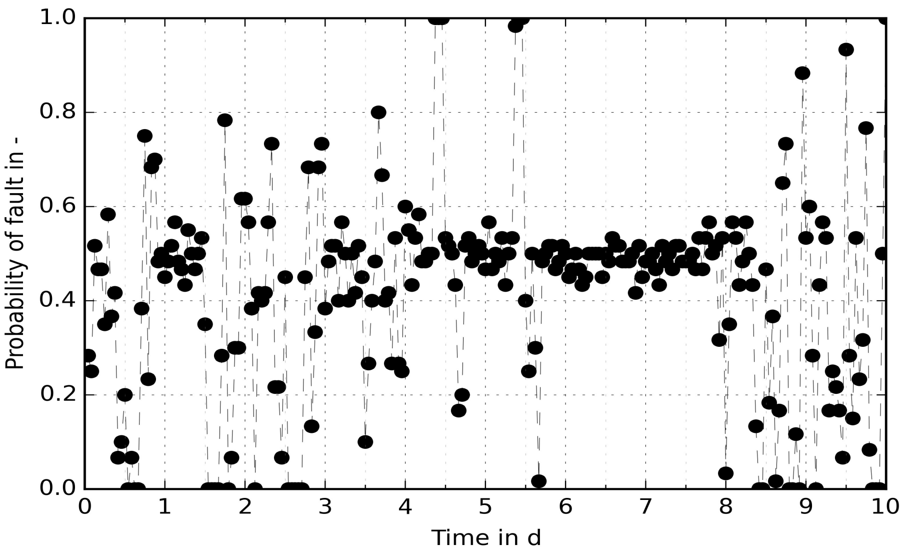

Figure 6: Reported probability of fault deploying rule-based analytic (eq. 1, ε_t =1.2) on hourly time intervals, while assuming a fully opened heating coil valve.

Discussion

The results show a demonstration of the intended workflow of the tool. However, some limitations have been identified.

In terms of scalability of the tool we are confident that it is applicable to use cases involving multiple buildings and analytics. Semantic web technologies and associated storage systems (triple stores) are well suited to host some millions of entries with acceptable performance (Dibowski 2013). For the part of analytics execution and ETL processing utilised technologies (Talend 2016) are scalable and may be deployed in a cloud with associated performance increases.

An unambiguous retrieval of information for the presented approach from standardised BIM formats such as the IFC remains difficult. A remedy may come from the development of specific exchanges, using the MVD (Model View Definitions) (buildingSMART 2016) mechanism.

Until now the capabilities to infer implicitly stated knowledge termed reasoning is utilised in a rudimentary manner for consistency checking and data integration. The tool could leverage on implicit knowledge generation in future, e.g. verify, crosscheck and supplement if necessary information from different information silos. An example is inferring the sensor typing from unit definition in BMS if BIM description is not complete.

CONCLUSION

In this work a software tool is presented which enables automated configuration of analytics for fault detection and diagnosis through providing contextual data from an ontology. A software architecture is presented to implement the necessary functionality to configure and deploy FDD analytics. For demonstration purposes, results obtained from implementing the tool are presented for the case of analysing real measurement data of an Air Handling Unit with rule-based analytics. For a test case application of nine days the tool detects faults using one rule and reports associated meta-data. Reports include spatial information such as location, technical affiliation of the sensor or actuator and period of time of the occurrence causing the analytic to report a fault. The information is obtained by linking a populated ifcOWL ontology with static BMS data in a single ontology.

The automation aspect resides in the initial design of a query and analytic pairing by experts. If this is done once and the specified analytic may be deployed and configured to arbitrary many systems under the assumption that static meta-information about the system is available.

Future work is related to further extend and revise the developed ontology. For this purpose it is intended to integrate and align with existing descriptions and standardisation efforts (Daniele et al. 2015, Ploennigs et al. 2012, Tomašević et al. 2015).

The tool could leverage on implicit knowledge generation in future, e.g. verify and crosscheck and supplement, if necessary, information from different information silos.

Future versions of the tool are intended to expand in their capabilities to access real time data for further testing the methodology.

ACKNOWLEDGEMENTS

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.

REFERENCES

AGEB 2014. Auswertungstabellen zur Energiebilanz Deutschland, AG Energiebilanzen e.V., Berlin Germany

Autodesk 2016. Revit 2016 Product Homepage. Accessed: 23/03/2016, http://www.autodesk.com

Berners-Lee, T., Hendler, J., Lassila, O. 2001. The Semantic Web. Scientific American 285, 28–37.

Bruton, K., Coakley, D., Donovan, P.O., Keane, M.M., O’Sullivan, D.T.J. 2013. Results from testing of a “cloud based” automated fault detection and diagnosis tool for AHU’s. in: ETFA, pp. 1–8.

Bruton, K., Raftery, P., Kennedy, B., Keane, M.M., O’Sullivan, D.T.J. 2014. Review of automated fault detection and diagnostic tools in air handling units. Energy Efficiency 1–17.

buildingSMART 2015. Industry Foundation Classes (IFC) 4 Add 1. Accessed: 23/03/2016, http://www.buildingsmart-tech.org/ifc/IFC4/Add1.

buildingSMART 2016. Model View Definitions. Accessed: 23/03/2016, http://www.buildingsmart-tech.org/specifications/ifc-view-definition/summary.

Corry, E., Pauwels, P., Hu, S., Keane, M., O’Donnell, J. 2015. A performance assessment ontology for the environmental and energy management of buildings. Automation in Constr. 57, 249–259.

Daniele, L., den Hartog, F., Roes, J. 2015. Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology, in: Cuel, R., Young, R. (Eds.), Formal Ontologies Meet Industry. Springer International Publishing, Cham, Switzerland, pp. 100–112.

de Wilde, P. 2014. The gap between predicted and measured energy performance of buildings: A framework for investigation. Automation in Construction 41, 40–49.

Dibowski, H. 2013. Semantischer Gerätebeschreibungsansatz für einen automatisierten Entwurf von Raumautomationssystemen. Diss. TU Dresden, Dresden, Germany.

DOE 2015. Long-term data on 3 office air-handling units. Accessed: 23/03/2016, https://trynthink.github.io/buildingsdatasets/

Domingues, P., Carreira, P., Vieira, R., Kastner, W. 2016. Building automation systems: Concepts and technology review. Computer Standards & Interfaces 45, 1–12.

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

Gruber, T.R. 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220.

Han, J., Jeong, Y.-K., Lee, I. 2015. A Rule-Based Ontology Reasoning System for Context-Aware Building Energy Management. in: CIT/IUCC/ DASC/PICOM, pp. 2134–2142.

Hitzler, P., Krötzsch, M., Rudolph, S. 2010. Foundations of Semantic Web technologies. CRC Press, Boca Raton, USA.

ISO 16484-5 2014. Building automation and control systems (BACS) – Part 5: Data communication protocol, International Organisation for Standardization ISO, Switzerland

Jena 2016. Apache Jena. Accessed: 23/03/2016, https://jena.apache.org.

Kofler, M.J., Reinisch, C., Kastner, W. 2012. A semantic representation of energy-related information in future smart homes. Energy and Buildings 47, 169–179.

Morbach, J., Wiesner, A., Marquardt, W. 2007. A Meta Model for the Design of Domain Ontologies, AVT-PT, Aachen, Germany.

Noy, N.F., McGuinness, D.L. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. Report, Stanford University, Stanford, USA.

Pauwels, P., Terkaj, W. 2016. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Automation in Construction 63, 100–133.

Pérez-Lombard, L., Ortiz, J., Pout, C. 2008. A Review on Buildings Energy Consumption Information. Energy and Buildings 40, 394–398.

Ploennigs, J., Hensel, B., Dibowski, H., Kabitzsch, K. 2012. BASont - A Modular, Adaptive Building Automation System Ontology. in: IECON 2012, pp. 4827–4833.

Protégé CoP 2016. Protégé – SWRLTabSyntax, Accessed: 23/03/2016, http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTabSyntax

R Core Team 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Schein, J. 2007. An Information Model for Building Automation Systems. Automation in Construction 16, 125–139.

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

Talend 2016. Talend Open Studio for Data Integration. Accessed: 23/03/2016, https://www.talend.com

Tomašević, N.M., Batić, M.Č., Blanes, L.M., Keane, M.M., Vraneš, S. 2015. Ontology-Based Facility Data Model for Energy Management. Advanced Engineering Informatics 29, 971–984.

Trojanova, J., Vass, J., Macek, K., Rojicek, J., Stluka, P. 2009. Fault Diagnosis of Air Handling Units. in: IFAC, Barcelona, Spain, pp. 366–371.

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

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. 2003b. A review of process fault detection and diagnosis: Part II. Computers & Chemical Engineering 27, 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K. 2003c. A review of process fault detection and diagnosis: Part III. Computers & Chemical Engineering 27, 327–346.

W3C 2016a. Resource Description Framework - RDF, Accessed: 23/03/2016, http://www.w3.org/RDF/.

W3C 2016b. Web Ontology Language – OWL, Accessed: 23/03/2016, http://www.w3.org/2001/sw/wiki/OWL.

W3C 2016c. SPARQL Query Language for RDF, Accessed: 23/03/2016, https://www.w3.org/TR/rdf-sparql-query/.

W3C 2016d. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, Accessed: 23/03/2016, https://www.w3.org/Submission/SWRL/

CESBP/BauSIM 2016 (Dresden, Germany)

I (we), the author(s) of the attached paper, have read the following Copyright Transfer and Disclaimer and agree to them by submitting the attached paper.

(1) The author(s) affirm(s), that the paper has not been published elsewhere and, if the paper is accepted, will not be published elsewhere prior to CESBP/BauSIM 2016.

(2) If the paper is accepted, the author(s) will automatically grant to the conference organizers a nonexclusive, royalty-free, perpetual, worldwide, irrevocable, sub-licensable and transferable license to publish the paper (in unmodified form) in any fashion (including but not limited to inclusion in the CESBP/BauSIM 2016 printed and CD proceedings, via electronic means such as the WWW, and inclusion in future compilations of papers). This "nonexclusive" license means that the author(s) are not restricted to future use of the material except that exclusive rights cannot be granted to another.

(3) The author(s) affirm(s) that he/she (they) has (have) the right to grant the license specified in (2), that is, publication by the conference organizers or its licensees will not be in conflict with the copyright or other restrictions.

(4) The author(s) acknowledge(s), that acceptance of the paper does not imply the conference organizers endorsement of or agreement with the ideas presented. Under no circumstances shall the conference organizers be liable for any damages resulting from used information included in the paper.

Figure 7: The author will be required to accept these conditions when they submit their paper via the web site.