# An algorithmic module toolkit to support quality management for building performance

Jan Mehnert<sup>1,\*</sup>, Dirk Reiß<sup>2</sup>, Stefan Plesser<sup>3</sup>, and Matthias Hannen<sup>4</sup>

<sup>1</sup>Institute for Building Services and Energy Design, TU Braunschweig, Mühlenpfordtstraße 23, 38106 Braunschweig, Germany <sup>2</sup>siz energie+, Mühlenpfordtstraße 23, 38106 Braunschweig, Germany <sup>3</sup>synavision GmbH, Welle 15, 33602 Bielefeld, Germany

<sup>4</sup>Chair of Software Engineering, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany

**Abstract.** Data from building automation systems is so far used for the operation of building systems and components only. The following work shows how this data can be used to enhance the building's performance by strategically detecting potential sources for building optimization. With this method, faults and optimization potentials of the building operation can be detected; thus, the quality gap regarding efficiency and comfort aspects between design and operation can then be reduced. Furthermore, the intelligent use of data enables the realization of economic savings to support facility management with regards to increasingly complex HVAC systems. Effective quality management – rapid, transparent and cost effective – is carried out with the aid of digital methods, which are already state of the art in other industries.

# 1 Introduction

Systems within buildings are increasing in complexity which is causing performance gaps during all phases of the life span of a building from design to operation [1] [2]. Faulty control sequences, inverted sensors, undefined setpoints etc. are causing a lack of the overall quality of the performance of buildings which often remains undetected. Applying active functional specification is a robust approach supporting the design of functions controlled by building automation systems (BAS) and in addition making them testable during commissioning and operation of a building [3].

The above-mentioned concepts are realized within a transdisciplinary team of building engineers from SIZ energie+ of TU Braunschweig, TU Munich, and software engineers from RWTH Aachen University. Additional partners from industry, such as Wilo SE and synavision GmbH, collaborate in a national research project funded by the Federal Ministry for Economic Affairs and Energy in Germany. Big Data from BAS is preprocessed within the software platform "Digital Performance Test Bench" by synavision GmbH [4]. The software platform is Fig. 1.

The data itself is stored in the Digital Performance Test Bench, which is shown on the upper left-hand side. It already has advanced capabilities to import and store additionally supported by component-based algorithmic modules for various tasks such as classification and semantic enrichment, but as well as automatic fault detection and pattern recognition using various data mining and artificial intelligence approaches. These modules form a foundation for flexible infrastructure to automatically analyze building performance. This contribution presents the first results of the project.

The paper is organized in the following manner: Section 2 describes an overview of the algorithmic module toolkit and provides a basic understanding of the concept. Section 3 and its subsections describe different applications of the toolkit in more detail, Section 4 concludes the paper and gives an outlook on shortcomings of and planned extensions to the concept.

## 2 Algorithmic module toolkit

The algorithmic module toolkit (AMT) presented in this paper consists of several components. An overview is depicted in

various formats of data and can run several preprocessing steps such as interpolation and converting measurement data to fit equidistant time steps. The algorithmic module toolkit itself consists of different stages that are shown as

<sup>\*</sup> Corresponding author: mehnert@igs.tu-bs.de

<sup>©</sup> The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

columns of the table on the right-hand side. Each step serves a certain purpose, such as importing data into the toolkit, preprocessing the data, analyzing the data, calculating scores for the executed analysis algorithms, and finally presenting the results to the client (e.g. in the Fig. 1.

In the example, concrete measured data in the form of comma separated values (CSV) is exported from the Performance Test Bench and then, in a first step, imported into the workflow of the AMT. Afterwards, the data goes through several preprocessing steps before different analyses are run. The final steps are the (semi-) automatic interpretation of the results and the reporting (here, in form of a PDF file). form of a PDF file or a webpage). The upper part of this figure represents the abstract infrastructure of modules that can be flexibly combined to complete workflows. The concrete instantiation of the toolkit modules is shown on the lower part of

The AMT itself is actually implemented in the R programming language [5]. Each module consists of well-defined input- and output interfaces so that they are freely combinable while serving a determined purpose. In the rest of this paper, concrete examples of module elements and application scenarios for them are described to illustrate the idea and possibilities of this approach.



Fig. 1: Structure and workflow of the algorithmic module toolkit

# 3 Results

Within the developed structure, each module has selfcontained functions that solve different tasks which are combined into a complete process that solves a certain problem. The following sections describe such processes and example module elements that are part of the algorithmic module toolkit.

# 3.1 Module Example 1: Data Type Detection

When analyzing data, the first step is to generate knowledge of the characteristic of each data point. Hence, data must be classified into classes and subclasses to allow a further analysis. A classification in subcategories of approximately 150 different data-point types is necessary so that typical HVAC systems can be tested. The information about data-point classes can be detected by two different methods. One is to filter each objects name given from the BAS. Here the difficulty lies in nonstandardized definitions or conventions. The second option is to use algorithms analyzing the trend of each time series to detect the above mentioned prior defined classes.

The ability to automatically detect major classes was tested by using time series collected from many AHUs. The classes were defined according to physical quantities such as temperature, pressure, volume flow as well as set points (0%-100%). Also, a class "other" containing datapoints which cannot be categorized is defined. With data from several AHUs in one building and algorithms from machine learning disciplines, the data was separated into a training and testing phase. In Fig 2 the vote of each so far unclassified type of data-point is shown. The algorithms which are used compares the extracted features from the testing phase which are known since they are to every single unknown time series from the testing phase. Hence, a vote is not the direct result of the class being assigned to a data-point but the mean of all votes for all individual time series which have been tested.



Fig 2: Percentage of votes of the base-case in which evaluates the robustness of the detected classes.

These votes then define which classes will be predicted and are compared to the prior defined real classes which have been trained by expert knowledge and are used as input by the algorithm. The results of this comparison are displayed in Table 1.

Table 1 : Confusion matrix of real trained and tested predicted

	clusses.								
teal class	Flow-rate					2	0		
	Set-point				2		0		
	Temp.			10			0		
	Pressure		2				0		
-	Other	2					0		
		Other	Pressure	Temp.	Set-point	Flow rate	Error ratio		
			Pred						

During another test-phase with AHU data from another building where the classes were known beforehand (expert knowledge), the votes from the applied algorithm were verified. These tests were conducted to explore the robustness of the algorithms.



Fig 3: Percentage of votes of the test phase with data from "unknown" times series which evaluates the robustness of the detected classes.

Even with lower votes, the classes can be predicted with a low error ratio since the algorithm acts in a democratic manner. Fehler! Verweisquelle konnte nicht gefunden werden. shows vote results in a confusion matrix of real and predicted classes with an error rate of 0.5 for the set-point which is predicted to contain the class "other".

Table 2: Confusion matrix of real trained data from one building and tested with data from a different building to predict classes.

Real class	Flow-rate					2	0
	Set-point	1			1		0.5
	Temp.			10			0
	Pressure		2				0
-	Other						0
		Other	Pressure	Temp.	Set-point	Flow rate	Error ratio

With the above shown method, prior trained classes can be classified according to their type, which is a prerequisite to further analyzing the data. Applying this approach in other instances, a detection of more detailed classes (e.g. differentiation between supply and exhaust temperatures, etc.) is possible.

### 3.2 Module Example 2: Tag repository

As the results of the data type detection need to be saved and made available in a proper manner, we developed a tag repository that is part of the infrastructure. It provides functionality in storing information about the data type that can then be used at a later point in the execution of the process. Here information such as site, discipline, system, component, position, sensor type, unit, etc. is attached. There are several ways to save this information. One way is to use simple tags to indicate that a data point has a certain type (e.g. adding a tag like temperature : sensor is a sign that a sensor is measuring the temperature). Another form of tags would be simple keyvalue-pairs, such as location : return flow, to save the fact that the sensor is measuring the return flow in a facility. Usually, tags are used in an additive manner, so that a data point is described by the sum of tags that are associated with it. In the given example, a temperature sensor that is installed in the return flow part of a system is described. One use for the saved information is described in Section 3.4

#### 3.3 Module Example 3: Outlier detection

From literature and standard codes, research was conducted to investigate minimum and maximum values according to the data-point type. Out of the above (Section 3.1) mentioned 150 relevant data point types for performance tests, 44 can be associated with thresholds from literature. When connected to a module which is able to detect the data-point type (e.g. from Module 1 in Section 3.1), faults in sensor readings such as offsets, connection failures, conversions, and default values can be detected. This module is used to assess data quality and if successful, to further trigger deeper analysis methods.

## 3.4 Module combination as a process

The algorithmic toolkit was established to be able to analyze various tasks. This approach has the benefit of using modules for different processes multiple times and configurations according to the requirements.

One application scenario of the AMT is the so-called Data Quality Check. It helps clients to quickly get an overview of the data that they retrieve from building systems (e.g. a BAS). The workflow is split into 3 levels which are shown in Fig 4.



Fig 4: Data Quality Check

First the data is exported from the Performance Test Bench, loaded into the AMT and processed accordingly. After running several analyses and calculating a number of parameters, reports are created as shown in the figure. This displayed information is split into two different parts. Some parameters can be calculated for any time series, regardless of the type of data point under inspection. Others need information about the type of a certain data point, which is where the tag repository is used to store all necessary information. This is used to compare the measured data under inspection with certain typical, average characteristics of data same types. For example, the maximum and minimum values or the distribution of values are compared to those of comparable data sets so that significant discrepancies can be highlighted.

The Data Quality Check itself is split into three different levels. The top level (Level 0) provides a very aggregated management overview of the data and reports in a traffic-light style about the quality of the data regarding different criteria. Furthermore, rankings of the data points are provided regarding various parameters (e.g. number of missing or invalid values in the data set) and linked to the more detailed views in the subsequent levels. Those (Level 1 and 2) show more information about each data point. This involves more and more detailed (statistical) factors as well as detailed figures that display the data itself and compare similar data points.

# 4 Conclusions and Outlook

With the AMT we established a robust infrastructure for the analysis of operational data from BAS. With its modules, complex problems are broken down to smaller components that can be (re-)used to compose processes which are able to automatically support performance testing within technical monitoring. With that approach the AMT supports the quality management of commissioning processes and supports the increase of the performance of buildings. The design approach of the toolkit is to ensure modules with a high reusability and the AMT being expandable to solve various tasks composed of different algorithms.

Besides the above described infrastructure of the AMT, several modules have already been implemented. The module data point classification (Secti on 3.1) defines the characteristics of each data point. With a training set of data from systems in one building, unknown classes can be detected automatically in another set. Knowledge about data point classes is essential for further investigations about the performance of buildings. With this gained semantics about data point types e.g.a time series being a supply air temperature further performance tests such as a suitable supply air characteristics depending on the outdoor air temperature can be derived.

Further developments in data point detection are foreseen so that many data points can be grouped or the detection of finer granularities of more detailed classes (e.g. differentiation between supply and exhaust temperatures etc.) is possible.

All gathered information is stored in the module tagrepository to make use of semantics (Section 3.2). The tag repository has the ability to store key-value information. This tag repository is then being used for handling information and as an interconnection of many modules as a process. The interconnectivity for example is shown in Section 3.4.

After the set-up of this robust infrastructure further modules will be deployed. These modules will support processes, among others, a detection of system states, an automated detection of faults within various HVAC systems and statistics of data point.

## Acknowledgment

This work is financially supported by the BMWi (German Federal Ministry for Economic Affairs and Energy) under the reference 03ET1426B.

## References

- P. de Wilde, 'The gap between predicted and measured energy performance of buildings: A framework for investigation', *Autom. Constr.*, vol. 41, pp. 40–49, May 2014.
- [2] C. van Dronkelaar, M. Dowson, C. Spataru, and D. Mumovic, 'A Review of the Regulatory Energy Performance Gap and Its Underlying Causes in Nondomestic Buildings', *Front. Mech. Eng.*, vol. 1, Jan. 2016.
- [3] S. Plesser, 'Aktive Funktionsbeschreibungen zur Planung und Überwachung des Betriebs von Gebäuden und Anlagen', Technische Universität Braunschweig, Braunschweig, 2013.
- [4] 'Perfect Building Performance synavision GmbH'.
  [Online]. Available: https://www.synavision.de/en/.
  [Accessed: 16-Nov-2018].
- [5] R Development Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2008.