# Data Integrity Checks for Building Automation and Control Systems

*Markus* Gwerder [a], *Reto* Marek [b], *Andreas* Melillo [c], *Maria* Husmann [a]

[a] Smart infrastructure, Siemens Switzerland Ltd., Zug, Switzerland, markus.gwerder@siemens.com.

[b] Center for integrated building technology, Lucerne school of engineering and architecture, Horw, Switzerland, reto.marek@hslu.ch.

[c] Institute of mechanical engineering and energy technology, Lucerne school of engineering and architecture, Horw, Switzerland, andreas.melillo@hslu.ch.

**Abstract.** Data from building automation and control systems are becoming more and more important, since they are used in a growing number of (novel and established) analytics applications such as fault detection & diagnostics (FDD), smart maintenance and optimization. However, the quality of such data is often poor due to erroneous installation, commissioning, data recording or meta-information. In addition, building automation engineering and service departments usually focus on implementing and maintaining basic control functionality – data acquisition, tagging quality, and analytics do often not take priority. Due to these data quality issues, a first important step in any data analytics operation is to ensure data integrity. One main goal of data integrity checks is to increase data reliability. The paper presents such checks for building automation applications, in particular three different types of plausibility checks for time series data: single signal tests, similarity tests, and reaction tests. Examples using data recorded from real building automation project are presented for each of the three check types, demonstrating the usefulness of these checks. Data integrity checks are set up and configured using the available metadata which – in our case – comes in the form of semantic models that are automatically generated from building automation engineering data. Many data integrity checks have been identified that are potentially of great benefit in practice – both as a stand-alone application or as first part in a data analytics process. The major prerequisite for successful data integrity checking is that the checks can be set up with minimal effort and executed periodically. To achieve a high degree of automation, semantic data is of great importance, because it is through them that the recorded time series are provided with context and meaning. The automatically generated semantic models from building automation engineering proved to be already rich in automation information and are sufficient for many of the checks investigated.

**Keywords.** Building automation, analytics, data integrity, data plausibility, semantic modeling.
**DOI**: https://doi.org/10.34641/clima.2022.271

## 1. Introduction

Data analytics and fault detection & diagnostics (FDD) methods are essential for the energy-efficient and comfortable operation of buildings. The objectives are manifold: Create transparency regarding optimizations in planning and operation, determine the origin of performance gaps, ensure and maintain the desired building performance, check the success of energy optimization measures, create reliable foundations for further optimization steps. However, these goals can only be achieved based on reliable and trustworthy data. The challenge here lies in the frequently poor quality of the data: incomplete, erroneous, unstandardized, or non-normalized data are quite common. Reasons for poor data quality originate from all different processes in building automation. Below, examples for such reasons per phase are given:

- Installation: faults in hardware, bad sensor placement, wiring errors
- Engineering: faults in control program, including faults in system integration; wrong or misleading names, tags, units
- Commissioning: point test not done properly, system test not done (e.g., hydraulic balancing)
- Operation: interruptions in connectivity or recording; gateway config.; neglected maintenance of building automation; changes in building automation software without re-commissioning and/or adaptation of management layer

Thus, ensuring data integrity is the first important step in data analytics and FDD. Data analytics – including the verification of data integrity – and its setup process can be significantly improved by combining measurement data (i.e., time series data) with semantic models, e.g., metadata about building geometries, building automation systems, components. Based on the knowledge contained within semantic models, data can be automatically checked for integrity, similar to what a human expert would do.

There are two key questions when assessing data integrity checks: (i) To what extent are the available semantic models applicable to set up and configure data integrity checks – both conceptually and in real-world project settings? (ii) Which data integrity check methods qualify through their broad applicability, high reliability and manageable computational needs for an implementation in practice.

There is a functional overlap between the data integrity checks investigated and widespread analytic solutions such as rule based FDD. However, the investigated methods are intended for specific tasks and strictly limited to data integrity verification, i.e., checking plausibility only. Checking for plausibility only has the distinct advantage that no prior knowledge is required about the building's usage and the correct design of its plants. This facilitates automatic testing – ideally, this can take place without any building- or plant-specific tuning. The main disadvantage of this approach is that the checks will not detect faults other than implausible behavior: the checks do not verify whether the systems operation is reasonable (e.g., inefficient system operation, unreasonable comfort setpoints, …).

Figure 1 shows a high-level overview of a data analytics workflow from «data collection», «merging» (unification and central storage), data pre-processing and data cleaning to subsequent checks. The checks considered in this paper are clearly distinguished from preceding classical checks regarding data acquisition, detection of data gaps and associated imputation, pre-processing, outlier detections and data cleaning. On the other hand, a separation shall be made to subsequent checks like e.g., checking design/dimensioning or comfort end energy by methods such as classical rule based FDD checks.

## 2. Methods

### *2.1 Processing and storing time series data*

Dealing with data from multiple sources can become complicated. Understanding the data, extracting, and transferring it to a central storage location is the first step in a data warehouse architecture [1]. Figure 2 represents a simplified representation of such a data workflow which includes as first steps the «data collection», «merging» and «pre-processing/data cleaning» of the data. Although the scope of this paper lies in the subsequent «data integrity checks», these first three steps are essential because each data integrity check as well as other subsequent checks such as e.g., FDD requires dedicated pre-processed data: Some methods rely on raw data, others rely on «resampled data» or «cleaned data» and further methods on «resampled and cleaned data».

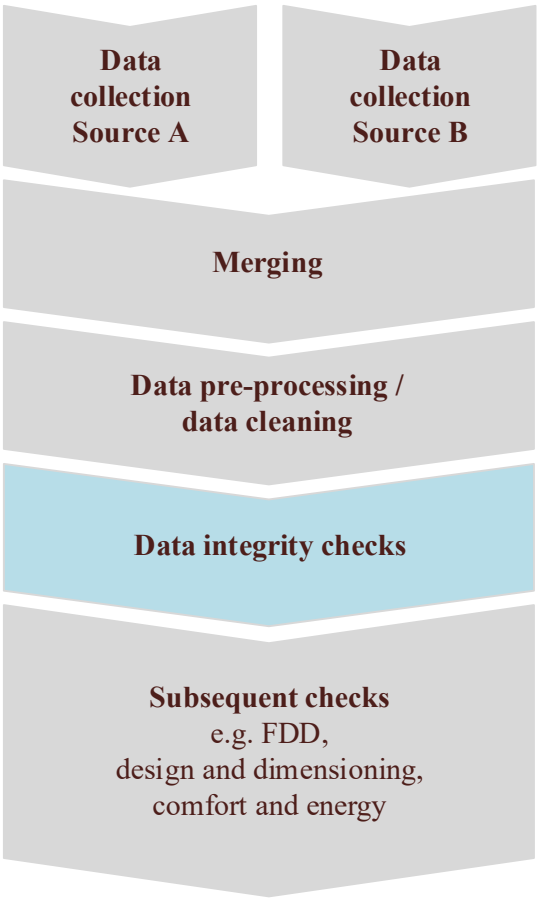


**Fig. 1 -** High-level overview of a data analytics workflow with incorporated data integrity checks.

Time series data processing steps such as correction/flagging of bad quality data or data resampling may profoundly affect the results of data integrity checks. Therefore, it is important to outline the different states of the data and the preprocessing steps that take place in between. Based on [1] and [2], a general data processing procedure is outlined in Figure 2, where each bucket represents a storage containing data in a particular state. The rectangular boxes in between represent the processing steps, including the activities required to transform the data.

### *2.2 Data check workflow*

Following the general data analytics workflow introduced above, data integrity checks as well as other checks can be performed using different kind of processed data (buckets). Figure 2 shows a possible data processing pipeline including checks of several types. Data integrity checks are colored blue, checks outside the paper's scope have a grey background. The data integrity checks investigated focus on plausibility using semantic information. The required semantic data is indicated in the yellow boxes.

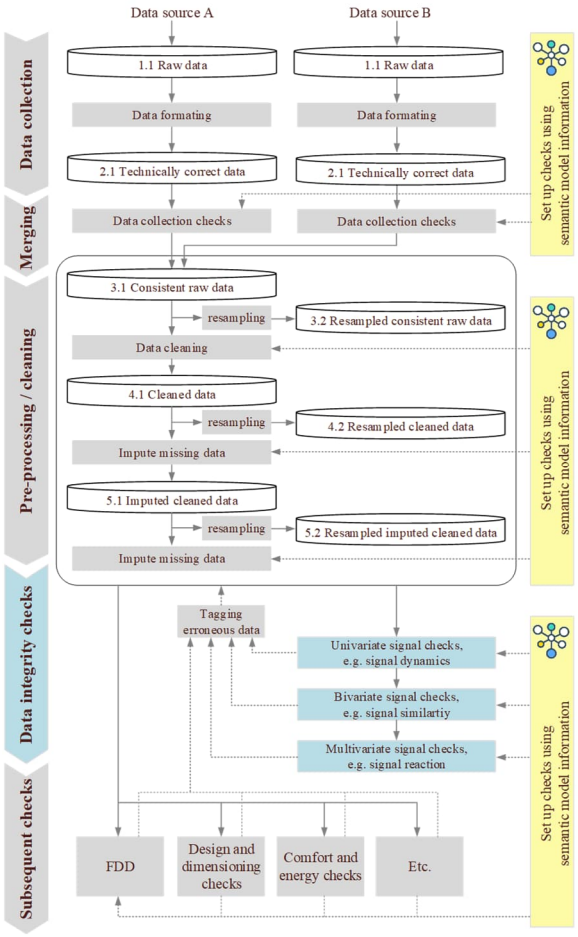


**Fig. 2 -** Data check workflow (blue boxes: checks considered in the paper, yellow boxes: semantic model information, grey boxes: checks outside the paper's scope, white buckets: data storages).

If a data integrity check fails, further analyses such as FDD do not make sense in general, as they would produce erroneous results. Failed data integrity checks should therefore tag erroneous data and subsequent checks should consequently evaluate such quality tags.

### *2.3 Development environment*

For prototyping and evaluation of data integrity check algorithms, a software test environment was designed and used for preprocessing and storing time series data, corresponding semantic model data access, and automated execution of data integrity checks. The software development environment consists of three main parts:

- Time series database: InfluxDB
- Graph database to store semantic information, also known as metadata: GraphDB
- Data workflow management platform to automate checks: Apache Airflow

Both databases and the data workflow management platform used to automate were put into operation on an Amazon Web Services (AWS). Time series data is exported daily from building automation and control systems (BACS) and stored in an AWS data lake. A script (run on an AWS server) processes and stores the pre-processed data. The pre-processed data corresponds to the bucket «4.2 Resampled cleaned data» in Figure 2. Another script ingests the data finally to the time series database.

The main hub is the workflow management platform. It performs periodic checks and stores and visualizes the results. The data for the checks are obtained directly from the two databases. The platform executes a chain of operations and analysis steps (data pipeline), where the output of an operation becomes the input to another [5].

### *2.4 Data sets*

Two different data sets representing building automation data from two buildings in Vienna were used for the project. These buildings are part of the ongoing research project «Aspern Smart City Research» (https://www.ascr.at/en/).

- Student home for 300 students
  The BACS includes room control, heat distribution (from district heating), ventilation as well as the electrical energy management of photovoltaic power plant and electric battery.
- Office building, ca. 8'000 square meters
  The BACS includes room control, heat and cold distribution and storage using a thermally activated building system, heat and cold generation using a heat pump, ventilation as well as the electrical and thermal energy management.

Since the buildings have been used in a large-scale research project already, a lot of data, knowledge, and information are available – much more than what is typically available in regular building automation projects. Therefore, data integrity check results can be applied and evaluated using multiple years of operational data and assessed in more detail based on the knowledge previously gathered.

### *2.5 Data integrity checks*

Most of the investigated data integrity tests were relatively simple rule-based methods based on statistical properties of the time series data, tailored to the considered application. All checks as well as data base access and basic visualization were implemented in Python. A statistical feature framework was developed for flexible and broad application of signal comparison operations. With customizable and extendable sets of conditions, statistical features, and actual tests, it is possible to design a custom data integrity check using the developed framework. The suggested procedure is divided in 4 steps which are summarized as follows:

1. Define conditions on data: Multiple conditions on multiple time series can be combined such that the timestamps, where all conditions are fulfilled, are flagged as valid. Consecutive valid

timestamps are forming valid ranges.
2. Calculate a list of selectable statistical features such as mean, standard deviation, polynomial fit parameters representing the valid time ranges (either calculate the features for each entire valid range or using a moving window within each valid range).
3. Use a methods collection to execute the appropriate test on the calculated features representation.
4. Visualize the outcome with the help of dedicated visualization methods.

### *2.6 Semantic Modeling of Building Technology and Automation*

Time series data can only be evaluated and analyzed if they are used in the context of the real plant. The data only becomes meaningful, understandable, and usable with so-called metadata. The goal is for the data to be self-describing, so that it can be used to add value without a great deal of manual effort. The intent of a semantic models for building automation and technology is to provide such information in a defined structured form so that it can be processed automatically by machines. The data check workflow presented in Figure 2 is an example of where semantic information can be used. Today, most building analytics applications are still mapped to the time series data (at least in part) manually, which is time consuming and error prone. Sematic models have the potential to improve this situation radically. Particularly important semantic model content for analytics applications is relational information such as supply chains, zones/command groups, locations, or control functional interactions (e.g., relations between controllers, control variables, setpoints, manipulated variables).

One approach to describe semantic information is to create ontologies (schemas) and describe instance data using these schemas. W3C provides standardized technologies such as RDF and OWL for that purpose and SPARQL for querying RDF data. There is no standardized ontology for the building automation domain, though several ontologies have been published by academia and various community groups [6, 7, 8]. We used our own ontology that – unlike the ones referenced – focuses on a functional description of the building automation system. The instance data can be exported from a building automation engineering tool or automatically generated from BACnet scans of systems that have been engineered with that tool. The building automation control application contains a lot of information relevant for data analytics (including data integrity checks). It is obvious that making this knowledge machine-readable and using it for analytics is highly advantageous compared to an analytics mapping process as described above.

Figure 3 shows a simple example of the functional model representing one part of a room thermostat (knowledge graph containing points, functions, locations, and relationships between them).

## 3. Results

In this section, a small extract of potential data integrity checks is given, as well as one concrete example check result for the three investigated test types.

### *3.1 Data integrity check collection*

Data integrity checks are preferably based on semantic models. While a comprehensive analysis of semantic data models is beyond the scope of this paper, the proposed approach is bottom-up: First, potential checks are considered which are meaningful and profitable. In a further step, it is analyzed which metadata are necessary for the check. Thus, a statement is then possible as to which data is missing in the available semantic models and which data is most important to be added from a practical point of view.

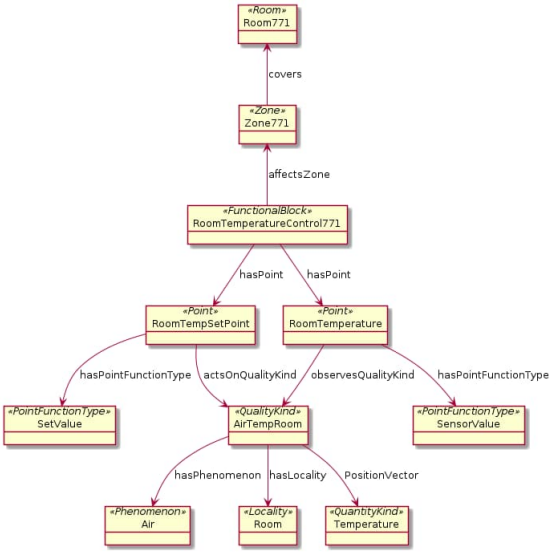


**Fig. 3** – Functional semantic model instance example.

Different data integrity checks for building automation data have been derived from literature (see [3], [4]) and developed by our own. Here, we provide a small extract of the collection in Table 1.

**Tab. 1 -** Extract of the data integrity check collection.

| Test type | Test description |
|---|---|
| Signal dynamic property | Minimal room air quality measurement over longer time periods should be close to outside air concentration. |
| Signal dynamic property | Room air quality measurements should exhibit a daily cycle when observed for a longer period of time. |
| Signal reaction | Room temperature measurement reacts on radiator valve position change when heat is provided by associated heat group. |
| Signal reaction | Room brightness sensor measurement reacts on light command/modulation change. |
| Signal reaction | Supply air temperature measurement reacts on heating coil valve position change when heat is provided by the associated |

| | |
|---|---|
| | heat group and air flow is provided by the associated air handling unit. |
| Signal similarity | Time series temperature measurements before and after air treatment steps in air handling units must be similar when air treatment is turned off and fan is running. |

### *3.4 Single signal check example*

Single signal checks are applied to one single measured / recorded time series. Additional information might be used such as geographical location and local time (e.g., to calculate solar position).

**Example**: Finding convenient locations for room temperature sensors is often not easy. The measurements might be influenced in an undesired way by internal or external heat gains. Single signal checks for room temperature measurements can detect such cases by assessing dynamic properties of the time series. For room temperatures, checking for unreasonable spikes is an adequate test.

Application of such tests to room temperature measurements of the office building introduced in 2.4 resulted in mostly passed checks. However, there were rooms which showed short positive spikes during 7-8pm local time in summer. An example time series of such a room temperature is given in Figure 4. Such spikes can lead – depending on the HVAC control – to unnecessary heavy cooling activity. To prevent this, either the sensor can be relocated, or the control program can be changed.

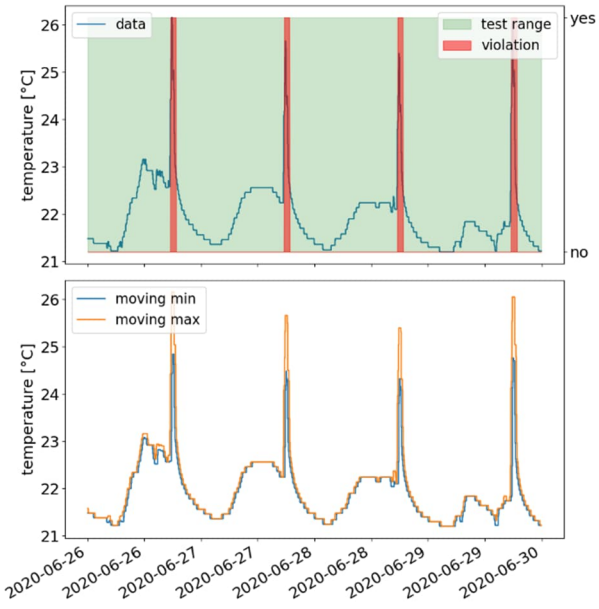


**Fig. 4 –** Example of failed room temperature single signal check. The difference between moving window minima and moving window maxima are used to identify peaks in the time series.

The assumed reason for the spikes (which was confirmed later) was exposure of the sensors to direct sunlight (because the blinds would have been controlled open in the evening). Using the semantic information of the room façade orientation, the assumption could also be consolidated (all the concerned rooms had western orientation). Even further consolidation is possible when additional measurements such as solar radiation or brightness is incorporated in the test (which of course makes the check no longer a single signal test).

### *3.3 Signal similarity check example*

Signal similarity tests check whether two time series are similar at certain points/ranges in time. They can be used to test signal similarities. In the simplest case, there are several sensors with which the same variable is measured. It can then be checked whether the signals are similar. In building automation, however, this case is typically rare, since redundancy is associated with additional costs. An example of such test methods is that of heat or cold meters, where the flow and return temperatures are measured by the meter as well as by separate sensors for control. In this case, the similarity only should be checked when the heat/cold fluid is circulating. Additional conditions for checking depend on the installation setup (e.g., heat meter on primary or secondary side).

In most cases, two signals are similar under specific conditions only, e.g., flow and return temperatures of a heat exchanger when no heat is transferred, or air temperature / humidity measurements before and after an inactive air treatment aggregate, such as temperature measurements before and after the heating coil.

**Example**: Redundant temperature measurements (heat meter & control temperature reading) in various heating circuits have been investigated based on data from the buildings introduced in 2.4. Figure 5 shows an example result for a flow temperature similarity test: The heat meter flow temperature measurement has a low resolution of 1 K and a relatively low sampling rate compared to the sensor used for control. Time periods where the similarity test passed are colored green, periods where the test failed are colored red. During all other time periods, the test was not applicable (i.e., conditions were not met). As can be seen, the example shows a high degree of similarity.

In practice, it can happen that heat meters are configured wrongly, e.g., the addresses of two heat meters are mixed up. Similarity tests can detect such misconfigurations. In the heating system of the school building (see 2.4), there are five main heat groups which are operated using similar schedules and setpoints. Nevertheless, the presented similarity test proved to be able to detect (artificially) misconfigured meters in all cases.

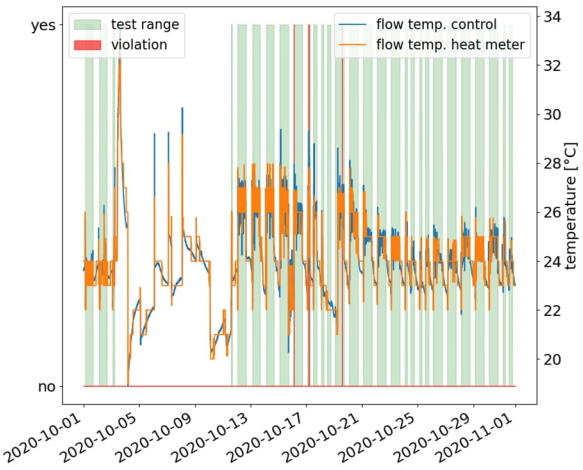


**Fig. 5 –** Example of passed heating circuit flow temperature similarity check. The test fails if the difference in the two signals is too large for a minimal amount of consecutive timesteps.

### *3.2 Reaction check example*

Signal reaction tests check whether a time series provides an expected response due to an event.

**Example**: A valve operation reaction test checks for plausible behavior regarding the operation of a temperature control valve / valve actuator in building automation. The checks are based on time series data of the valve command, temperature readings and the operating state of the associated pump. The check passes if the signals correspond to what is defined as plausible; it fails if this is not the case. Reasons for the check to fail:

- The valve is stuck (open, close, mid position)
- There is a fault in one of the sensors or sensor installation used to measure the required data
- There is a fault in the pump associated to the valve
- There is a fault in the measurement processing chain (including data recording)
- There is a fault in the mapping or labeling of the data

The checks are assumed to be particularly helpful for valves/valve actuators without position feedback signal. Ideally, the check can be applied broadly to different types of valves and applications using typically available time series data. Very common applications are heating or cooling mixing circuits, but also room temperature control valves.

A practical room automation example of a failed test is shown in Figure 6. In this case, a concrete core conditioning system is operated by opening and closing heating and cooling valves – the data comes from the office building, see 2.4. The figure shows a few days in August 2020 (cooling season). Time periods with opened cooling valve are indicated by green areas. The cooling command marked by dashed red lines was identified as failed by the applied reaction test:

Despite the opening command of the valve (and flow temperature provided by the cooling circuit sufficiently low), there is no reaction visible in the concrete core temperature. The test result proved to be correct. The reason was found to be a fault in the commanding of the valve actuator by the room controller leading to sporadically non-executed commands. The fault could be fixed by a software update of the room controller.

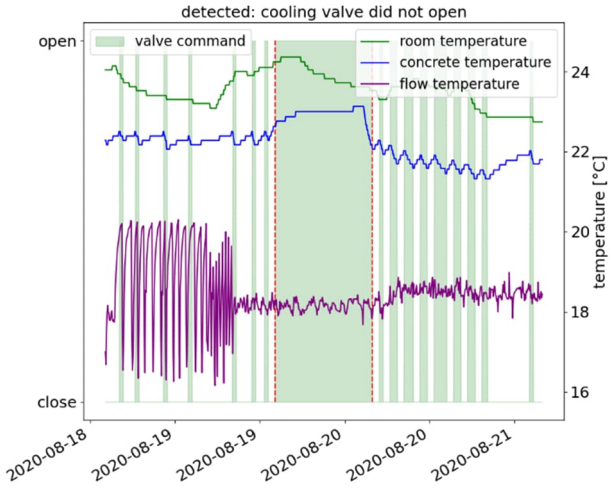


**Fig. 6 –** Example of failed cooling valve reaction check.

## 4. Conclusions

Data integrity checks are of great benefit to enhance the quality of subsequent data analyses and increase confidence in the data (as well as the plant operation the data reflects). These checks can be considered as the first part in a data analytics process. But they can even be beneficial as stand-alone tests.

There are several data integrity checks that are considered promising in practice. However, a thorough evaluation of their performance would require applying the checks to many more (labeled) data sets than has been possible. Most of the investigated / developed methods are potentially broadly applicable (different buildings, control application, ...) and can be performed from the beginning of building operation, which means they can be useful already in the commissioning phase. No training with historical data is needed. The challenges in the practical application of the checks lie mostly in the robust and broadly applicable design of the test methods. Currently, e.g., a reaction test must be configured for the expected reaction speed (which cannot be easily derived from semantic information).

The main prerequisite for successful application of data integrity checks is that they can be set up with minimal effort and executed periodically. In order to achieve a high degree of automation in setting up the checks, semantic data is of great importance, because it is only through them that the recorded time series data are given context and meaning. The semantic models we used were automatically generated from

the building automation control solution. Therefore, these models are already rich in automation information. Currently not yet contained aspects in these models are some cross-plant relationships, particularly supply chain relationships, which are used in many of the data integrity checks studied. Fortunately, in many cases, such relationships are contained in the control program to implement demand-driven control, which means this information can be added to the semantic models. Other missing aspects such as hydraulic topology or equipment specification typically are not needed for control and therefore would have to be incorporated using other data sources such as BIM. However, many of the checks studied can be set up and executed without this additional information. Furthermore, semantic models that include control functions are also useful for subsequent analytics applications [9].

## 5. Outlook

The results stimulate the development and application of «plausibility check» type data integrity tests. With the valuable knowledge and experience gained, a systematic process will be followed in the future: Based on the complemented and evaluated integrity checks collection, checks with the highest potential benefits will be selected for further analysis. The development of the selected checks will then focus on broad and robust application. Automation, reporting and visualization of the checks are then prototyped, and successful methods will be applied outside of limited test data sets.

## 6. Acknowledgement

The authors would like to gratefully acknowledge the funding by Siemens Switzerland Ltd. The authors would also like to thank Prof. Olivier Steiner and Prof. Philipp Schütz (both Lucerne school of engineering and architecture) for their support.

## 7. References

[1] Kimball R., Ross, M. The data warehouse toolkit: the definitive guide to dimensional modeling. 3rd Edition. John Wiley & Sons. 2013.

[2] De Jonge, E., van der Loo, M. An introduction to data cleaning with R. The Hague: Statistics Netherlands. 2013.

[3] Brady, N., Lloyd, R. Trust the raw data? The importance of applying data integrity. CLIMA 2016, 22-25 May 2016, Aalborg, Denmark.

[4] Gitzel, R. Data quality in time series data: An experience report. CBI 2016, 29 Aug - 1 Sep 2016, Paris, France.

[5] Wang, E., Cook, D., Hyndman, R. J. A new tidy data structure to support exploration and modeling of temporal data. Journal of Computational and Graphical Statistics. 2020;29(3):466–478.

[6] Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Berges, M., Culler, D., Gupta, R., Kjærgaard, M.B., Srivastava, M., Whitehouse, K. Brick: Towards a Unified Metadata Schema for Buildings. BuildSys '16. Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments. 2016.

[7] Hammar, K., Wallin, E.O., Karlberg, P., Hälleberg, D. The RealEstateCore Ontology. In: Ghidini C. et al. (eds) The Semantic Web – ISWC 2019. ISWC 2019. Lecture Notes in Computer Science, vol 11779. Springer, Cham. 2019.

[8] Project Haystack. http://project-haystack.org/. 2021.

[9] Ramanathan, G., Husmann, M., Niedermeier, C., Vicari, N., Garcia, K., Mayer, S. Assisting automated fault detection and diagnostics in building automation through semantic description of functions and process data. BuildSys '21: Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. 2021.

### *Data Statement*

The datasets generated during and/or analysed during the current study are not available due to confidentiality agreements, but the authors will make every reasonable effort to publish them in near future.