Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 219 (2023) 416–421

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022

IoT-based Diagnostic Assistance for Energy Optimization of Air Conditioning Facilities

Benjamin Nast*, Achim Reiz, Kurt Sandkuhl

Rostock University, 18051 Rostock, Germany.

Abstract

The Internet of Things (IoT) is a significant trend in the field of information technology and encourages the implementation of cyber-physical systems, smart connected products, and new business models. Many enterprises struggle to create business value from IoT technology because they have difficulties defining their organizational integration. Model-driven engineering (MDE) is considered an effective technique to address the complexity of IoT application development. Existing approaches focus on requirements from a technical perspective and exhibit a lack of integration with organizational and business model aspects. The paper proposes a modeling approach and a tool to support the development and configuration of IoT solutions in the example of air conditioning and cleanroom technologies (ACT) facilities. The main contributions of this paper are (a) an architecture for the IoT application, (b) a modeling language and tool support for IoT modeling, and (c) the expected practical benefits in the industrial case.

Peer-review under responsibility of the scientific committee of the CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2022

Keywords: Internet of Things; IoT; Digital Transformation; IT Architecture; Model-Driven Engineering; Modeling Methodologies.

* Corresponding author. Tel.: +49 381 498 7409
E-mail address: benjamin.nast@uni-rostock.de

Peer-review under responsibility of the scientific committee of the CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2022
10.1016/j.procs.2023.01.307

1. Introduction

The Internet of Things (IoT) is a significant topic in research and industry and contributes to Digital Transformations (DT) in many domains [1]. IoT technologies support the implementation of cyber-physical systems, smart connected products, and new business models. Ordinary resources are transformed into intelligent objects that can sense, act, and behave [2]. This new industrial environment is supported by tools and applications capable of developing intelligent systems that can perform autonomous tasks. Those systems are empowered by Artificial Intelligence (AI) [3], including reinforcement learning and machine learning (ML). Combining the IoT with data science capabilities enables decision-making that optimally impacts physical systems, thus increasing the relevance of predictive maintenance [4]. The concepts of IoT and AI are therefore very suitable to face the challenges of Industry 4.0.

Today, many approaches support the specification, design, and implementation of IoT solutions [5], from a technological and development perspective. However, enterprises struggle to create business value from IoT technology or hesitate to invest in IoT efforts because they have difficulties defining organizational integration [6]. Model-driven engineering (MDE) is considered a promising technology in the domain of cyber-physical systems [7]. Methods for MDE should encompass organizational and system development and integration. Existing model-based approaches also focus on requirements from a technical perspective and exhibit a lack of integration with daily processes that they ought to support [5].

We propose a modeling language and tool support to compensate for this lack. The work described in this paper is based on an industrial case from air conditioning and cleanroom technologies (ACT). The case study company is a medium-sized enterprise from Germany that designs, develops, installs, and operates industrial-sized ACT facilities. The goal of the project is to optimize energy consumption and build a basis for predictive maintenance. Therefore, additional sensors and control systems must be integrated into ACT facilities and connected to a network. The resulting IoT solution provides diagnostic support for potential optimizations in such facilities and the operational processes of the case study company.

This work proposes an architecture and a modeling tool to support system development and integration. The main contributions of this work are (a) an architecture for the IoT application, (b) a modeling language and tool support for IoT modeling, and (c) the expected practical benefits in the industrial case.

The paper is structured as follows: Section 2 describes the architecture of the IoT application. Section 3 introduces the developed tool and modeling language, followed by an overview of the practical help the solution will offer the ACT technicians. The paper concludes with the current state of the research and future work.

2. An Architecture for Using Models to Design IoT Solutions

Industrial air conditioning systems are not produced on scale, but designed individually for every client. Many components need to interact in these systems, and we need a variety of sensors to understand what is happening inside them. However, the signals from these sensors are not interpretable out of the box. Often, they only measure an analog signal from 0 to 10 volt, which needs to be converted to a physical measurement unit. To make sense of these measurement points and understand the performance of the system, they then need to be processed and set in context with each other. Further, as the configurations of these air conditioning systems differ, so do the arrangement of measurements. Only some systems have a cooling unit, and some use them just for cooling, while others also use the cooling unit for dehumidifying.

These heterogeneous structures impact the IT architecture as well. On the one hand, the systems are heterogeneous in their configurations, and the data needs heavy preprocessing to be usable. On the other hand, much potential is in comparing systems with similar configurations. At the same time, the implemented system must be usable by the air conditioning experts. This emphasizes a system that does not require many IoT- and analysis-specific skills but is usable by the end user. Fig. 1 presents the planned architecture.

The application is divided into two views: Physical and logical. The former configures and manages the sensors. It is responsible for connecting the wireless Lora sensors and gateways to the server stack. Lora is a wireless low-power network protocol for transmitting data. The data is encrypted and sent to a gateway, which is itself connected to a Lora server in the backend. The Lora server has the decryption keys and can relay the data to another service [8].

In our case, the Lora server (Fig. 1: *Lora*) receives the data in a binary format representing an analog signal from 0 to 10 volt. Afterwards, the data is sent to a second serverside application (Fig. 1: *prep*) to process the sensor's voltage signal into a physical unit. It has the necessary calibration and interpretation information to convert the signal to the given measured aspects, e.g., *humidity*, *temperature*, or *percentual value opening*, and also adds information on the location and type of the sensor.

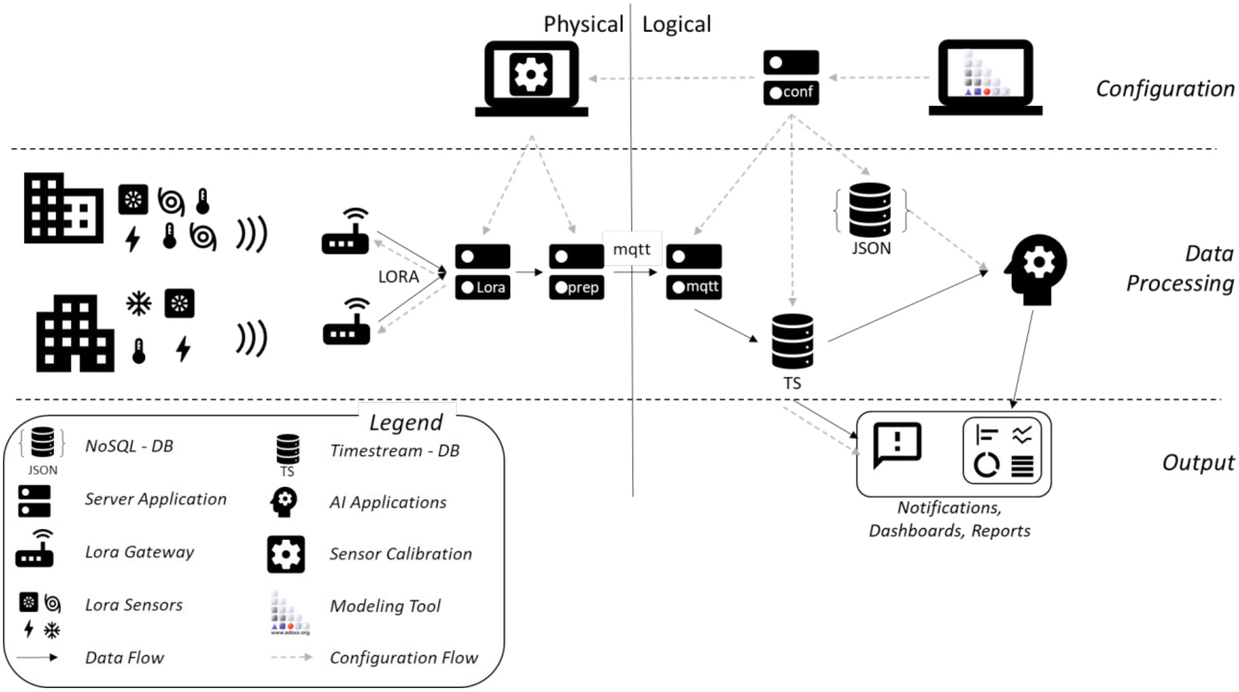

Fig. 1. Architecture of Model-Driven System.

The now interpreted signal is sent to the logical layer via mqtt. This layer is responsible for making sense of the data, providing the processing capabilities, and generating the outputs. The data arrives at an mqtt broker (Fig. 1: *mqtt*) with an unambiguous id of the corresponding system, the kind of sensor location, and its measurement. Afterward the data is transported to a time-series database (Fig. 1: *TS*), providing persistent storage. This persistent storage builds the foundation for generating simple reports and analyses and more sophisticated reports based on AI-based applications (more on that in the upcoming section). As the measured systems are heterogenous, a given machine learning system cannot compare them as a single group. A document-centric database (Fig. 1: *JSON*) contains the necessary background information to understand, group, and compare the air conditioning systems that are similar. This background information also determines the additional processing required to interpret the data.

All the different components for a new system need to be configured initially. Setting up each component manually would burden a lot of the complexity on the air conditioning engineer, thus requiring much costly training and limiting the application scenarios of the solution. We solve that issue by providing a configuration layer with two separate applications. The first entry point is the ADOxx-based modeling tool as further described in the upcoming section. The tool gathers the required information, packs it into a JSON file, and sends it to a server application (Fig. 1: *conf*). This server application configures the processing layer:

Regarding the logical view, the conf-application tells the mqtt broker the new topics on where new data is expected, prepares the TS database, including the visualizations, and stores the static master data of the new system in the JSON database. The conf-tool also manages the alignment of the logical and physical stack by assigning unambiguous ids for the components. The configuration of the physical layer itself requires a separate tool. This tool assigns the Lora gateways and sensors to the server stack, gathers the sensor type (e.g., *temperature*, *humidity*), and connects the sensor values to the alignment ids, thus configuring the mqtt topic and bridging the logical and physical perspective. Further, it allows sensor testing and calibration by configuring offsets to incoming values.

3. Modelling Language and Tool Support

To support the ACT technician in designing the solution, we developed a modeling language and tool support using the ADOxx meta-modeling platform. It is a development and configuration platform for the implementation of modeling methods. This platform enables the realization of full-fledged modeling software that contains procedures and functionalities in the form of mechanisms and algorithms in addition to the modeling language.

Our modeling language for ACT facilities follows the requirements of the case study. Fig. 2 shows a model of a fictitious facility as an example. The classes of the meta-model represent the possible components in such facilities. Attributes contain, for example, information about the type of the components or the unique ids of sensors. The symbols are based, among other things, on the standard DIN EN 12792 [9] and the knowledge of domain experts involved in the project. The graphical representation, as well as the designation of the components, depends on the entered attributes. Arrows show the logical relationship between the components and the airflow direction. Sensors are represented with a circle and a dash to the component or place in the model. The letters in the circle indicate the sensor type, and the measuring unit is shown below. An example model (see Fig. 2) shows some of the objects of the modeling language and contains six sensors: Two each for controlling the supply and exhaust air (Temperature in °C and CO_2 content in ppm). Further, the two air filters are equipped with a sensor for the pressure difference (unit of measurement: Pascal).

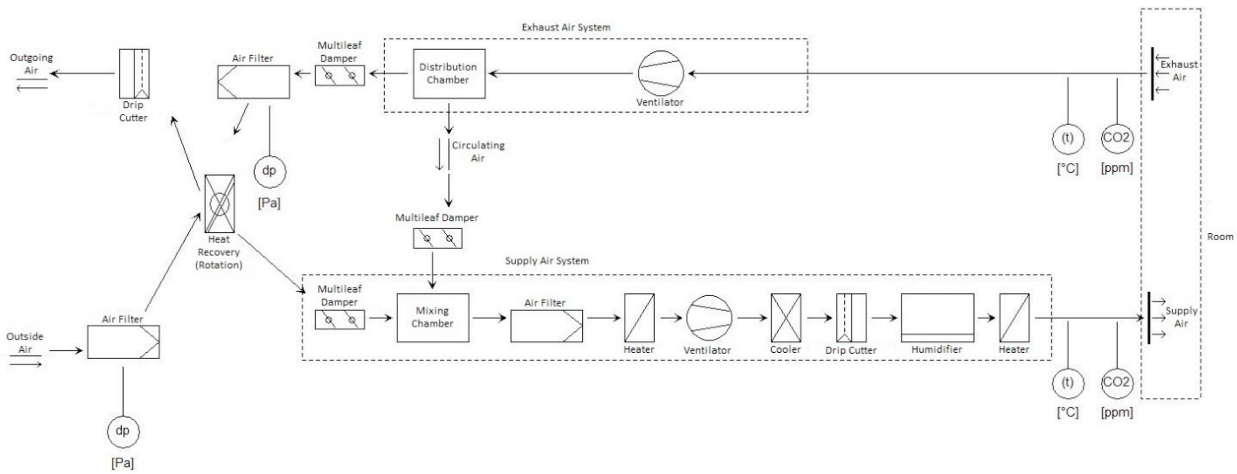

Fig. 2. Objects of the Modeling Language.

Creating a new facility in the tool first facilitates the entering of master data. Dialog boxes query the required information about the location, the operator and owner, and the components of a facility. This input automatically creates the respective modeling objects, including the attribute values (e.g., type of the heat recovery or size of the ventilators), and thus supports the creation of the model. After an object is placed in the model, the technician can assign unique ids to its corresponding sensors. The tool allows exporting data in different ways for different purposes.

On the one hand, information about the master data of a facility and the ids of the sensors is converted into a JSON file and sent to a server application using an HTTP request as described in Section 2. On the other hand, a documentation of the created model can be exported. These can be used for presentation or internal purposes of the case study company. This documentation can, for example, help make the customer understand his facility, the results of energetic inspections, or even the need for renewal and repairs.

The developed tool enables the technician to configure installed sensors. Parts of the physical configuration are covered by gathering hardware-related sensor information and mappings to measured aspects. The data processing of the logical part is enabled using the master data to understand which kind of ACT system is used. As a result, the user does not need specific knowledge or skills in IoT and data analysis.

Fig. 3 shows an example of an existing ACT facility of the case study company. In the next step, it can be used to evaluate the suitability for supporting the development and configuration of the IoT solution.

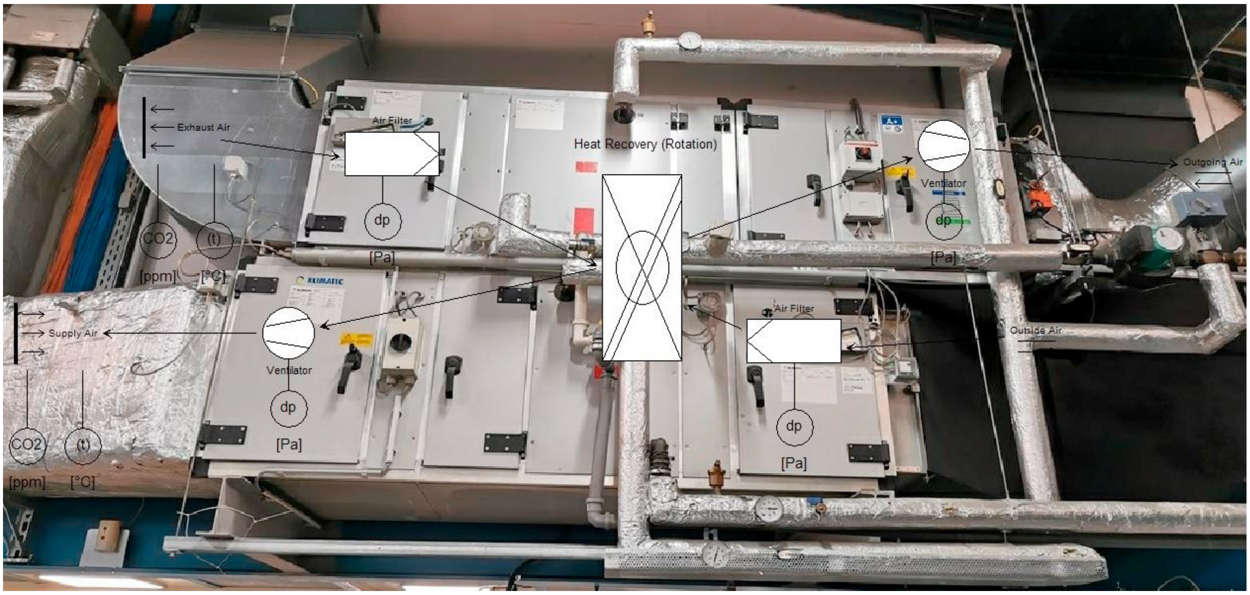

Fig. 3. Model Example of an ACT Facility.

4. Bringing the Application to Use

Our system's overall goal is to reduce energy consumption in industrial-sized air conditioning systems. Due to regulatory requirements and possible warranty issues, our system does not actively interfere with the ACT control systems but informs a technician on possible wasteful settings or wrongful system behavior. It is then up to the staff to take corrective measures.

The gathered information gets to the technician either using event-triggered notifications (E-Mail, SMS) or through the consumption of automatically or semi-automatically generated reports. While the former shall allow the quick identification of malfunctions and irregular behavior, the latter aims to evaluate the system's performance in the long term. It can improve cybernetics, assess the system's dimension, or draw conclusions based on comparing similar air conditioning systems.

Currently, the (in Germany) mandatory inspections of the industry-scale air conditioning systems are performed manually by a visiting technician. The technician inspects the current behavior of the system and the wearing parts. Some facilities have a building control system installed, which, at times, if appropriately configured, can provide a historical sensor readout. However, as our industrial partner tells us, the available data (if any) is mostly too incomplete to evaluate the system's performance for almost all of the currently installed air conditioning systems.

In this regard, the first starting point of the new analysis system is relatively trivial dashboarding and alarming functionalities. A simple historical view of the airflow, e.g., can reveal whether the air conditioning system shuts down properly in the evening and off-week. Similarly, an alarm can be configured for such a scenario to inform the technician automatically. Scenarios like these occur, e.g., through a manual interference in the timing mechanism for an event in the evening. While the solution to such a scenario is trivial from a technical perspective and can be built simply on top of the time-series database, it can quickly reveal significant saving possibilities.

Over time, the number of connected facilities will increase, thus also contributing to a rising pool of analyzable data. In combination with the configuration settings on the installed systems, it shall allow us to compare similar air conditioning systems. We expect it to reveal hidden overconsumption of energy and guide improvement recommendations for existing systems. Further, using ML technologies, we plan to detect changing behavior automatically using anomalies in the data to enable energy savings beyond the manual human analysis of historical graphs.

5. Conclusion

Connecting physical devices and analyzing their telemetry data can leverage massive potential for efficiency gains. Especially ACT systems require a vast amount of energy while often not being digitalized adequately. At the same time, the ACT staff that operates the given facilities has no computer science background. The personnel needs an interface with a familiar design language.

This paper presented an architecture and interface for connecting ACT systems with the Internet of Things. The underlying IT architecture can get rather complex because the sensor values cannot be used out of the box but need heavy processing. The partition into a physical and logical view splits the system's complexity into two separate, manageable tasks. The two tools for configuring the system (especially the modeling software) hide this complexity from the end-user. Thus, we expect that our solution requires little additional training for the ACT technician. The research on this project is still ongoing. While we finalized the presented architecture and modeling tool and connected a first prototypical ACT site with sensors, the subsequent steps are implementing the processing capabilities and the automatic configuration procedure.

Acknowledgements

Work presented in this paper was partly funded by grant KIDIRA from Federal State of Mecklenburg-Vorpommern, Germany, and European Union Funds.

References

[1] E.B. Hansen, S. Bøgh, Artificial intelligence and internet of things in small and medium-sized enterprises: A survey, Journal of Manufacturing Systems 58 (2021) 362–372. https://doi.org/10.1016/j.jmsy.2020.08.009.

[2] R.Y. Zhong, X. Xu, E. Klotz, S.T. Newman, Intelligent Manufacturing in the Context of Industry 4.0: A Review, Engineering 3 (2017) 616–630. https://doi.org/10.1016/J.ENG.2017.05.015.

[3] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, H. Vincent Poor, Federated Learning for Internet of Things: A Comprehensive Survey, IEEE Commun. Surv. Tutorials 23 (2021) 1622–1658. https://doi.org/10.1109/COMST.2021.3075439.

[4] M. Compare, P. Baraldi, E. Zio, Challenges to IoT-Enabled Predictive Maintenance for Industry 4.0, IEEE Internet Things J. 7 (2020) 4585–4597. https://doi.org/10.1109/JIOT.2019.2957029.

[5] G. Fortino, C. Savaglio, G. Spezzano, M. Zhou, Internet of Things as System of Systems: A Review of Methodologies, Frameworks, Platforms, and Tools, IEEE Trans. Syst. Man Cybern, Syst. 51 (2021) 223–236. https://doi.org/10.1109/TSMC.2020.3042898.

[6] Kurt Sandkuhl, Ulf Seigerroth, Digital Transformation of Enterprises: Case Studies and Transformation Paths, in: Doug Vogel, Kathy Ning Shen, Pan Shan Ling, M. N. Ravishankar, Jacky Xi Zhang (Eds.), 25th Pacific Asia Conference on Information Systems, PACIS 2021, Virtual Event / Dubai, UAE, July 12-14, 2021, 2021, p. 35.

[7] A.W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, S. Yin, Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution, EEE Ind. Electron. Mag. 11 (2017) 6–16. https://doi.org/10.1109/MIE.2017.2648857.

[8] S. Devalal, A. Karthikeyan, LoRa Technology - An Overview, in: Proceedings of the Second International Conference on Electronics, Communication and Aerospace Technology (ICECA 2018): 29-31 March 2018, IEEE, Piscataway, NJ, 2018, pp. 284–290.

[9] DIN EN 12792:2004-01, Lüftung von Gebäuden_- Symbole, Terminologie und graphische Symbole; Deutsche Fassung EN_12792:2003, Beuth Verlag GmbH, Berlin.