Plug & Play Retrofitting Approach for Data Integration to the Cloud

1st Santosh Kumar Panda
inIT - Institute Industrial IT
Technische Hochschule Ostwestfalen-Lippe
Lemgo, Germany
santosh.panda@th-owl.de

2nd Lukasz Wisniewski
inIT - Institute Industrial IT
Technische Hochschule Ostwestfalen-Lippe
Lemgo, Germany
lukasz.wisniewski@th-owl.de

3rd Marco Ehrlich
rt-solutions.de GmbH
Industrial Security
Cologne, Germany
ehrlich@rt-solutions.de

4th Mainak Majumder
inIT - Institute Industrial IT
Technische Hochschule Ostwestfalen-Lippe
Lemgo, Germany
mainak.majumder@th-owl.de

5th Jürgen Jasperneite
Industrial Automation Branch
Fraunhofer IOSB-INA
Lemgo, Germany
juergen.jasperneite@iosb-ina.fraunhofer.de

***Abstract*—Driven by rapid digitalisation, production systems are becoming more flexible and adaptable with the help of emerging concepts like the Internet of Things (IoT) and Industry 4.0. Often, these transformations are not fully implemented in Small and Medium-sized Enterprises (SMEs) due to the replacement cost of existing machines. This paper aims to develop a Plug & Play retrofitting platform, where Industry 4.0 compliant sensor systems can be attached, detected, and configured automatically to the existing production environment. The purpose of the retrofitting is to integrate the sensor system with a cloud platform that would provide persistent storage for sensor data as well as the functionalities to perform monitoring, analysis, and predictive learning.**

***Index Terms*—Industry 4.0, Retrofitting, Sensor Node, Plug & Play, AutomationML, eCl@ss, OPC UA, Asset Administration Shell, Amazon Web Services**

I. Introduction

With the emergence of Industry 4.0, the industrial environment is transforming with an increasing number of machines and their inter-connectivity due to rising levels of digitalisation and automation [1], [2]. Flexible and adaptable production is becoming more of a norm in the industry with the introduction of Cyber-Physical Systems (CPSs) and access to real-time data. However, the implementation of Industry 4.0 concepts is still not often realised to its full potential in SMEs due to the replacement cost of existing hardware [3]. One of the approaches could be to retrofit an existing industrial environment with an Industry 4.0 compliant independent system. Industry 4.0 is a combination of modern technologies such as IoT, cloud computing, big data, augmented reality, artificial intelligence, and information security. The retrofitting approach doesn't mean to use all available Industry 4.0 technologies but by identifying processes in participant SMEs, features, and requirements that are required for retrofitting.

This paper intends to define a retrofitting approach by developing a platform where sensor systems can be easily attached, detected, configured, and integrated with the existing environment for monitoring, data analysis, and predictive learning purposes. The Plug & Play deployment of such approaches should configure and integrate itself with minimum overhead similar to a Universal Serial Bus (USB) device when connected to a Personal Computer (PC). Further, the Plug & Play deployment requires to have a standardised semantic description, communication capabilities, and persistent data storage [4]. Nowadays, various cloud platform provides the functionality for storage, analysis, condition monitoring, and predictive learning based on the acquired data [5]. Therefore, the sensor data is integrated into a cloud platform to avail these above features. Such an approach will provide Industry 4.0 functionalities to legacy machines, and the Plug & Play approach will allow faster connection and increase efficiency without the need for highly trained engineers.

The remainder of the paper is organised as per the following. Section II gives an insight into the related work for Plug & Play, retrofitting approach, and cloud integration. Section III briefly describes AutomationML as the semantic device description. OPC Unified Architecture and its discovery features are described in section IV as middleware communication. Section V details the requirement for the retrofitting approach. Section VI provides an approach for Plug & Play data integration between the sensor systems and the cloud platform. The implementation is evaluated in Section VII. Finally, Section VIII concludes the paper by summarising the approach for Plug & Play integration and detailing the future scope.

978-1-7281-5297-4/20/$31.00 ©2020 IEEE

II. RELATED WORK

Microsoft introduced Plug & Play concept for the computer system in Windows 95. It was the first operating system which would attempt to detect and configure USB devices automatically after connecting them to a computer. Universal Plug & Play (UPnP) allows detecting specific services on a network automatically for home and office network [6]. Analogous to the Plug & Play for computers, Industry 4.0 also applies the concept of Plug & Produce capabilities for production systems, where new entities can be integrated into an existing system with minimum resources and short time resulting in production flexibility.

The white paper [4] details a use-case for Plug & Produce integration of field devices into an existing system. It proposes an approach based on OPC Unified Architecture (IEC 62541) mapped with AAS based on standardised device description such as OPC UA Device Interface (IEC 62541-100) [7], NAMUR NE 131[1], AutomationML (IEC 62714), and Field Device Integration (IEC 62769). M. Schleipen et al. describe an approach based on OPC UA and AutomationML for Plug & Work and details a prototypical implementation in [8]. S. Profanter et al. define a Plug & Produce device discovery and auto-configuration by implementing OPC UA multi-cast discovery services for an industrial network in [9]. S.K. Panda et al. describe an automatic approach to integrating the digital representation of components based on OPC UA [2].

As part of the Industry 4.0 vision, the future industry is going to be highly interconnected with modern sensor systems and communication infrastructure. Machines and field devices should be adaptable and self-optimised by sharing data properties and configuration details among one another. However, it is not the usual case for older machinery and equipment because the replacement costs for modernisation of the industry are very high. An alternative can be a retrofitting of legacy machines by developing communication interfaces and IoT devices to provide Industry 4.0 advantages without buying new machines.

C. Horn and J. Krüger describe a retrofitting approach for machinery with legacy communication interfaces in [10]. They propose an application-oriented concept utilizing different connecting approaches to enable new value-added services to interact with machinery with legacy connection interfaces. In [11], H. Haskamp et al. summarise the process for retrofitting a flexible manufacturing system by integrating legacy Programmable Logic Controllers, Radio Frequency Identification, and commercial cloud technology based on an Industry 4.0 compliant digitalisation of components with OPC UA interfaces. In [12], T. Lins et al. propose a retrofitting approach through a platform regardless of the model or type of the industrial equipment and offers resources to integrate this equipment with Industry 4.0. The platform utilises technologies such as Software-Defined Networking (SDN), OPC UA, and cloud computing by establishing integration and communication management of industrial equipment and IoT sensors.

Cloud Computing can be referred to as the on-demand delivery of computing power, data storage, applications, and other IT resources through a cloud services platform via the Internet or local network in case of public or private cloud respectively [13]. One of the key benefits of cloud computing is the opportunity to replace infrastructure expenses, such as hardware and maintenance, with a low variable cost that scale with the business requirements. With the increasing popularity, several models and deployment strategies have emerged to meet the specific needs of different users and provides different levels of control, flexibility, and management.

In [14], the authors discuss the changes in data integration approaches for enterprise applications around the use of cloud-based systems. Also, the authors define various new patterns or requirements that enterprises should consider when they approach data integration within cloud-based domains. C. Zhu et al. focus on the integration of Wireless Sensor Network (WSN) and cloud computing in [15]. The authors have identified four research problems for the WSN and proposed working solution for each one of them. Also, the authors in [10] and [12] describe the cloud integration of machinery based on retrofitting approach.

However, it can be observed that Plug & Play, retrofitting, and cloud computing can be brought together to have a flexible, fast, cost-effective, and scalable Industry 4.0 application. Therefore, this paper combines all these approaches and proposes a solution for Plug & Play retrofitted data acquisition system that is integrated with a cloud platform to perform various application-oriented functionalities. Moreover, the work also utilises the Industry 4.0 technologies such as Plug & Play, Asset Administration Shell, cloud integration to provide digitalisation through retrofitting.

III. AUTOMATIONML

A. Overview

AutomationML is a data exchange format developed by AutomationML e.v. in 2006 that not only focuses on process and plant engineering but also covers all the relevant information regarding production systems [16]. It is an appropriate candidate as a data exchange model for heterogeneous manufacturing industry due to its open and vendor-neutral XML-based format. It follows an object-oriented paradigm to model physical and logical components of any production system. It is also standardised within IEC 62714 standard series [16]. AutomationML describes the system hierarchy based on Computer Aided Engineering Exchange (CAEX). CAEX is an XML based neutral data format which allows storing hierarchical object information, e.g., structure of a production system. AutomationML follows a modular structure to accommodate other existing XML-based formats for data exchange under a single umbrella through the use of CAEX for modelling of production system components, its properties, and relations. AutomationML provides integration with COLLADA, PLCopen, eCl@ss, and many other formats.

[1]https://www.namur.net/en/index.html

AutomationML is divided into four parts based on the CAEX format and described briefly in Table I [16].

TABLE I
AUTOMATIONML COMPONENTS

Name	Description
Role Class Library	Comprises of *RoleClass* that describes the semantics of objects in an abstract manner.
System Unit Class Library	Stores the reusable system components or known as templates for system modelling specified by *SystemUnit Class*.
Interface Class Library	Consists of external interfaces among objects specified by *InterfaceClass*
Instance Hierarchy	Represents the actual engineering data following an object oriented and hierarchical structure through internal elements referencing role and system unit classes to provide their semantics.

B. Asset Administration Shell model in AutomationML

The concept of the Asset Administration Shell (AAS) described in the Reference Architecture Model for Industry 4.0 (RAMI 4.0) [17] implements the methodology to represent a physical device virtually and its technical capabilities. It provides a standardised interface to describe its data elements through a semantic description and the communication capabilities through a middleware communication protocol. The combination of the physical device, its AAS, and the communication capabilities between them is defined as an Industry 4.0 component. The retrofitted system is considered as an asset and designed based on AAS standards. The AAS will be able to provide all the properties of its embodied asset and the current and historical data for its connected sensors. The AAS consists of various submodules. The submodules are responsible for the identification, data properties, or services based on a standardised format. In this paper, the retrofitted system components are modelled by considering AAS as per the standardisation provided in [18].

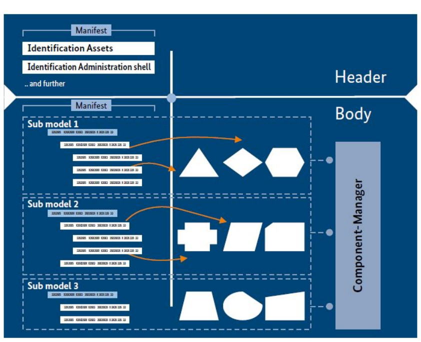

Fig. 1. Structure of an Asset Administration Shell [19].

The modelling of the retrofitted system must follow the semantics defined in the Asset Administration Shell representation by using AutomationML. The AAS consists of the embodied asset as well as the submodels to describe the properties and elements of the asset. Optionally, it can contain views and dictionaries for stakeholders and semantics respectively. The dictionaries contain concept descriptions that are used to describe the semantics of the submodels. The submodels consist of several submodel elements, and these elements can be properties, variables, or collection. These submodels and their elements can adhere to an industry-specific catalogue that defines the syntax and semantics for the components and their properties. One of such catalogue can be eCl@ss[2] which can exchange data and is interpretable by all the participants in a production system through its object definition hierarchy with unique object properties [20]. eCl@ss is an ISO/IEC-compliant data standard for grouping of product-specific properties to provide a hierarchical and semantic representation of products and materials through its numeric class structure [20]. These classes feature lists of standardised properties with an accurate description for subsequent identification of products and services. The submodel elements can refer to an eCl@ss property through its *semanticId* in *HasSemantics*. The AML meta-model provides various *RoleClasses* that defines the elements of the AAS as well as the mapping to create the *InstanceHierarchy* in AutomationML. Fig.2 shows the AAS meta-model *RoleClasses* in AutomationML editor.

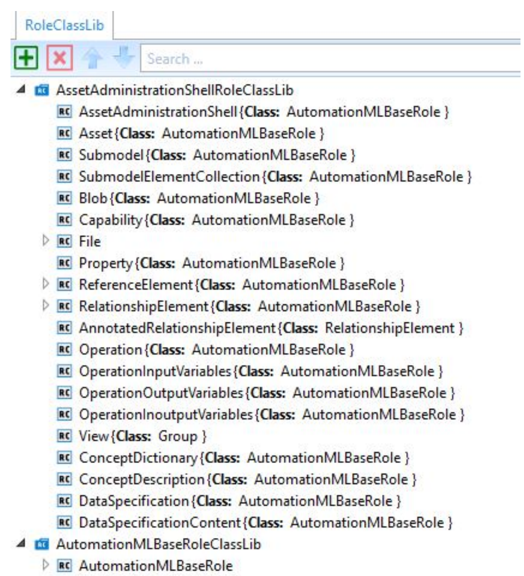

Fig. 2. Asset Administration Shell meta-model represented through AutomationML RoleClass [20].

IV. OPC UNIFIED ARCHITECTURE

A. OPC UA in Industry 4.0

OPC Unified Architecture (OPC UA) is a service-oriented, platform-independent, machine-to-machine, and machine-to-enterprise communication protocol developed by the OPC Foundation [21], [22]. OPC UA defines the communication mechanism through a client-server or a publisher-subscriber

[2]https://www.eclasscontent.com

model and allows information to be connected in different ways by extending additional vendor-specific information to the OPC UA base model. One of the challenges in Industry 4.0 is to provide a secure, standardised exchange of data and information between devices, machines, and services across different industries. OPC UA can be deemed as the appropriate communication solution for Industry 4.0 as it provides various functionalities, such as simple data acquisition, monitoring, control and analysis, security, and integrating other standards through its companion specification by providing standardised information model.

B. AutomationML to OPC UA Specification

As described in section III, AutomationML is an XML-based description for production systems. It can describe the system components ranging from small sensors to various complex machines, including their hierarchies and topologies. This information can be exchanged among parties through a communication middleware. Therefore, OPC Foundation and AutomationML consortium have worked together to create an OPC UA companion specification for AutomationML [23]. The information described in AutomationML can be communicated using OPC UA architecture by combining AutomationML and OPC UA. The companion specification [23] provides all the necessary information regarding the transformation rules to generate OPC UA information model from an AutomationML model.

C. OPC UA Discovery

The discovery process in OPC UA allows clients to find servers in the same network and then discover how to connect to the server. The individual servers must be registered to a discovery server which provides a way for clients to find the registered servers. The different discovery servers in OPC UA are Local Discovery Server (LDS), Local Discovery Server with Multicast Extension (LDS-ME), and Global Discovery Server (GDS) [24]. For a Plug & Play setup, the underlying devices must be identified without any specific pre-configuration and must be discovered automatically. For this purpose, OPC UA Discovery provides the necessary functionalities through its services to have a Plug & Play setup.

V. Requirements Analysis

The sensor systems are connected to an edge device which is responsible for integrating the sensor data into a cloud platform. The edge device is also responsible for discovery feature and communication capability from the local device to the cloud platform. The sensor system must execute the necessary steps to be able to provide the retrofitted system properties and the connected sensor data through a standard information model. Also, it must be discovered by the edge device automatically to advertise its information and connection requirements. The requirements are identified based on the primary task of this paper and are divided into four sections – Sensor Node, Discovery of Sensor Node, Auto-Configuration, and Cloud Integration.

A. Requirement for Sensor Node

The sensor node is a hardware system that has to be designed in such a way that it can be easily adapted to different industrial application scenarios. Depending on the individual process or machine characteristics, different sensors are connected to the hardware through serial communication to measure different physical process parameters. To be able to achieve Plug & Play functionality, the sensor node must be able to set up all necessary functionalities, such as device information, communication capabilities, and self-advertisement abilities without any manual intervention. To increase equipment efficiency, fast connectivity, and reduce cost, the sensor nodes are needed to be connected and start operation without any specific pre-configuration. The following requirements have been derived for the sensor node to accomplish an effortless retrofitting.

1) Each sensor node must receive the network connectivity without any manual intervention through any modern IP based wireless connectivity.
2) The sensor node must be capable to provide the data elements of the retrofitted system as well as the connected sensors through a standardised information model from a semantic description e.g. AutomationML.
3) The sensor node should setup its communication capability through a platform independent middleware to provide its data elements across the existing system for cloud integration.
4) Each newly connected sensor node should be advertised across the existing system to be discovered and configured to provide all sensor information.

B. Discovery of Sensor Node

After the sensor node is plugged into the existing network, it needs to perform the tasks described in the previous section to be able to register itself with the edge device. In an IP based network, automatic network connectivity can be assigned through the Dynamic Host Configuration Protocol (DHCP). Therefore, the edge device must function as a DHCP server as well as must have a discovery service to discover the underlying sensor nodes and can connect to them without any network-specific pre-configuration. The discovery service not only should have the information of registered servers when they are connected but also it should the functionality to remove the information of sensor nodes when they are plugged out of the system.

C. Auto-Configuration

The auto-configuration is divided into 2 parts – auto-configuration in the sensor node and auto-configuration in the edge device.

1) Auto-Configuration in the Sensor Node: The configuration in the sensor node should be carried out by identifying the semantic description, creating its information model from the semantic description to provide access to its data elements, and creating the necessary communication stack to communicate with its sensors. The whole process should be done automatically to achieve the Plug & Play functionality.

2) Auto-Configuration in the Edge Device: The primary challenge for the edge device is to identify the sensor nodes, configure itself to communicate with them, and perform necessary tasks for each sensor node for the cloud integration. The edge device should be able to access the information model of each sensor nodes to create configuration files for cloud integration. Since one of the main focus is to avoid pre-configuration, a Plug & Play process should be developed to complete these tasks automatically by incorporating OPC UA discovery client and cloud functionalities.

D. Cloud Integration

A cloud platform has to be chosen that has the functionality to provide cloud-based management features like analytics, persistent storage, and machine learning models. OPC UA is being used for the necessary communication functionalities for sensor nodes and the edge device for the Plug & Play integration. Therefore, the cloud platform should also have the functionality to connect to an OPC UA server. Based on these above requirements, Amazon Web Services (AWS) has been used for the cloud application as it fulfils these basic cloud requirements for the retrofit approach.

AWS IoT Greengrass extends cloud capability to local devices by using *AWS Lambda* functions to create serverless applications. these applications can be deployed in local devices in order to establish a connection to *AWS IoT Core* as well as to perform specific tasks. In this case, each *AWS Lambda* function consists of an OPC UA client and the *AWS IoT Greengrass* functionalities. It is responsible for connecting to its OPC UA server and establish communication between the edge device and *AWS IoT Core*. Further, *AWS IoT SiteWise* can be used to collect the sensor data to monitor it in large scale. With *AWS IoT SiteWise*, sensor data streams can be grouped from multiple locations by production line and facilities so operators can better understand and improve processes across facilities. *AWS IoT SiteWise* can create models of industrial processes and equipment across multiple facilities, and then automatically discover and visualise live and historical asset data through customisable charts and dashboards through launching a web application. The AAS meta-model can be mapped to asset models in *AWS IoT SiteWise*, and the AAS model can be instantiated from the asset models. The modelling can adhere to the AAS specification and the sensor values can be assigned to the asset properties in *AWS IoT SiteWise* to provide real-time and historical data for the sensors.

VI. Plug & Play Retrofitting Approach

The Plug & Play approach defines an architecture to integrate the sensor nodes to the cloud. The retrofitting architecture is divided into three layers – Sensor Node layer, Edge Device layer, Cloud layer. Also, the approach details the implementation and visualisation of AAS for the sensor nodes. Fig. 3 shows the general architecture for the retrofitting approach.

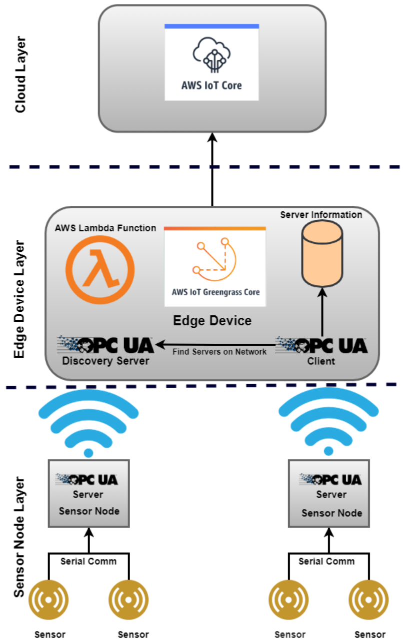

Fig. 3. Proposed retrofitting architecture.

A. Asset Administration Shell for Sensor Nodes

In the design phase, the AutomationML data model is considered as the AAS model which consists of the Instance Hierarchies for the AAS and the *ConceptDescription*. The model uses eCl@ss International Registration Data Identifier (IRDI) as the semantic identifier for the submodels and submodel elements. The retrofitted system is considered as an asset, which is embodied inside the AAS. The sensors of the newly installed sensor system are regarded as the *OperationalData* submodel. The *OperationalData* submodel contains the sensors as *InternalElements*, which is capable to provide the run-time data in operational phase through its attributes. The OPC UA *AddressSpace* derived from the AutomationML model is considered as the Administration Shell in the operation phase with live-data. An example of a bakery machine is shown in Fig. 4, which is retrofitted with temperature and pressure sensors.

B. Sensor Node Layer

A retrofitted system can have only one sensor node. However, the sensor node layer consists of one or more sensor nodes, each comprising of multiple numbers of sensors across the shop floor for multiple retrofitted system. These nodes acquire the IP address from the DHCP server of the edge device by connecting to it via an IP based wireless protocol such as WiFi or 6LowPAN (IPV6 over Low Wireless PAN). The sensor nodes are preferably based on a Micro-Controller Unit (MCU) or any single board computers, such as Raspberry Pi or Arduino. The sensor nodes are designed by analysing

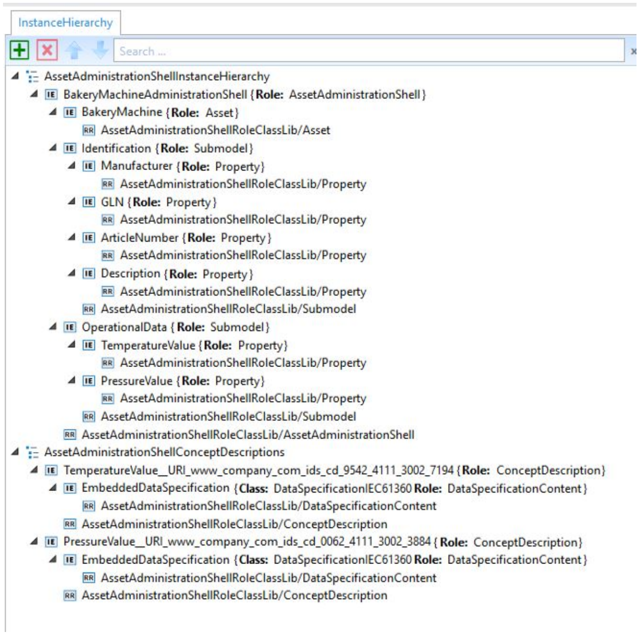

Fig. 4. Asset Administration Shell for a bakery machine based on AutomationML and AAS meta model.

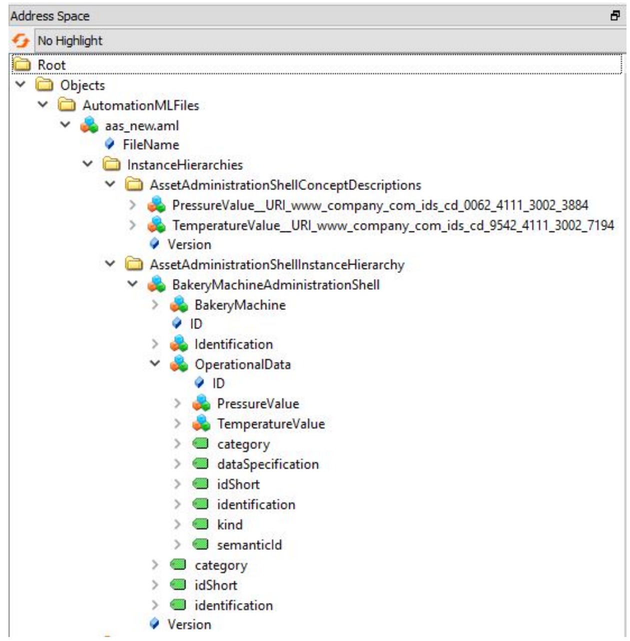

Fig. 5. OPC UA AddressSpace of the Asset Administration Shell for a bakery machine.

the processes and machines involved in the existing system to identify important process parameters based on their influence on each process and product. After the influencing process parameters were identified, corresponding sensors according to the identified parameters are connected to their respective sensor node via serial communication such as Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI), or Modbus. An *AutomationML* model based on AAS specification is stored in the sensor node for the retrofitted system. The sensor node provides information of the retrofitted machine through its *AddressSpace* and also the communication capability to fetch the values from these sensors through a callback. Once the server is set up and operational, it registers itself to the discovery server of the edge device. Fig.5 shows the OPC UA *AddressSpace* of the AAS in UAExpert.

C. Edge Device Layer

This layer consists of a device which can be an embedded Linux device or an industrial PC. The device has an OPC UA discovery server that provides the connection requirements for each OPC UA server from the underlying sensor nodes through OPC UA Discovery services. It also has an OPC UA client which fetches the endpoints of the registered servers and connects to them to get the information from their respective *AddressSpaces*. It stores the server details in a database. Based on this information from the database, an *AWS Lambda* function is created for each sensor node. The function provides the sensor values by creating a JSON payload and sends it to the cloud via the *AWS IoT Greengrass core* software which manages the communication between the device and the cloud application.

D. Cloud Layer

The cloud layer consists of the *AWS IoT Core*, which is responsible for providing necessary administrative functionalities. Such functionalities can be security and access certificates for the communication between the edge device and the AWS cloud. The *AWS IoT Core* manages the *AWS IoT greengrass Group* which deploys programming models to the edge device through the associated *AWS Lambda* function. Each edge device has its *AWS greengrass Group*. And, each associated sensor node of the edge device has its *AWS Lambda* function. These functions are associated with the corresponding *AWS greengrass Group* of the edge device. Also, the AAS model hierarchy is created in the cloud application, which represents its corresponding sensor node and the sensors. In *AWS IoT SiteWise*, the asset models and assets are created for the AAS meta-model and its instances respectively based on the AAS. And, they are being configured to get the sensor values from *AWS IoT Core* and store the real-time and historical values of their respective sensor. Fig.6 shows the sequences for the Plug & Play retrofitting working principle.

VII. Evaluation and Implementation

The purpose of this work was to achieve a retrofitting platform where a sensor system can be capable to acquire process parameters in legacy machines and integrate them to a cloud platform for analysis. Aiming to achieve Industry 4.0 functionalities, a case-study was defined for food industries where the quality of ingredients would be analysed based on the data collected by the retrofitted system. Therefore, a temperature/pressure sensor system is designed for a bakery machine as most of the industrial food processes require pre-

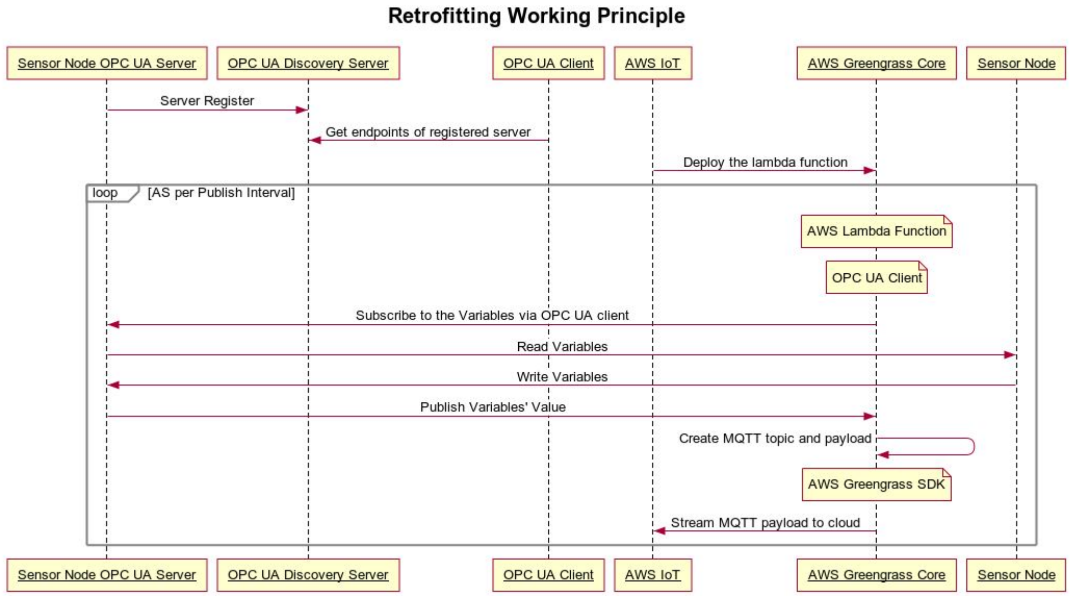

Fig. 6. Retrofitting working principle sequence diagram.

cise temperature analysis being one of the important physical parameters and can influence various chemical reactions in a food manufacturing industry.

The Plug & Play set up in both the sensor node and the edge device is implemented in C++. The sensor node comprises an STM32 MCU coupled with a WiFi module. The sensors are connected to it through the I2C protocol. The AAS model is created by using AAS Package Explorer, and exported to an *AutomationML* model. The generated model referred to eCl@ss for the semantics of submodels and submodel elements. The OPC UA implementation is done by using Open62541 stack, which also provides the discovery services. Once the sensor node is set up, it acquires an IP address from the DHCP server of the edge device and creates the OPC UA information model by accessing the AutomationML model for the AAS. An AutomationML to OPC UA transformation tool is developed based on the companion specification. Then, it creates the OPC UA server configuration files based on the information model and I2C functions.

The edge device is a Raspberry Pi enabled with a wired internet connection. A DHCP server is implemented in the Raspberry Pi and it has been checked that the sensor node is assigned with an IP address from the Raspberry Pi. An LDS server is implemented and evaluated that it can discover all the registered servers. The Plug & Play setup in this device is responsible for fetching all the information of the registered servers and store them in a database. Then, it connects to the database and access the *AddressSpace* of the registered servers and create the AWS Lambda configuration files and deploy them into the *AWS IoT Cloud*.

In *AWS IoT SiteWise*, asset models and assets are created for the AAS meta-models and instances. In *AWS IoT Core*, rules have been created for each subscriptions from the AWS Lambda function. Then a visualisation portal in *AWS IoT SiteWise* is created according to the requirements to show the sensor values. The Fig. 7 shows the values of a temperature sensor connected to retrofitted bakery machine.

VIII. CONCLUSION

The idea of this paper is to develop a platform where sensor systems can easily be attached, detected, and automatically configured into an existing system for monitoring, analysis, and learning purpose. Therefore, a sensor system is designed and integrated with a cloud platform in a Plug & Play setup in order to acquire sensor data for the above purposes. The requirements were discussed, and respective technologies were mapped with each of them. An architecture has been proposed for the Plug & Play retrofitting approach comprising of 3 layers – Sensor Node Layer, Edge Device Layer, and Cloud Layer based on *AutomationML* to OPC UA transformation, *OPC UA Discovery*, and *AWS* respectively. The working principles have been implemented in C++ and have been tested and evaluated. The system can easily be integrated with any of the existing systems for process monitoring and analysis also the acquired data can further be used for a learning model to identify defect by using *AWS IoT Analytics* and *AWS IoT Greengrass ML Inference* respectively.

Further, the idea is to implement each sensor node with other low power wireless protocols and be compared with WiFi with respect to performance and power consumption. Further, the sensor nodes will be integrated with different serial communication to make it heterogeneous and provide an interoperable solution through OPC UA. Also, the AAS for the

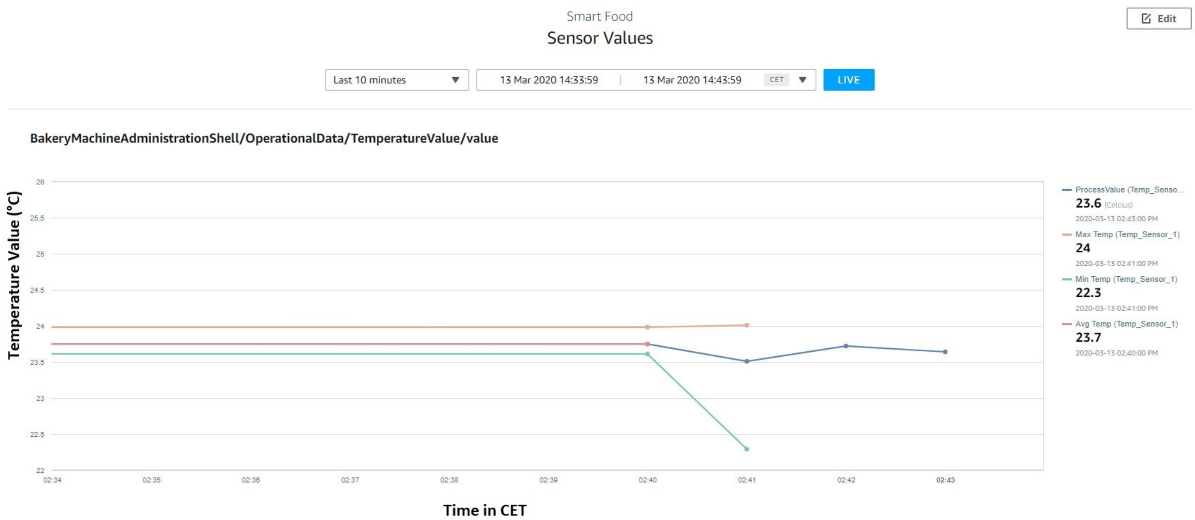

Fig. 7. Temperature Sensor Value in AWS.

retrofitted system will be modelled according to a composite Industry 4.0 component by including the information from the installed sensor system.

Acknowledgement

This work is supported under the Smart Food Technology-OWL initiative together with partners from small and medium enterprises, industry companies, and research institutes from OWL University of Applied Science and Arts– Institute Industrial IT (inIT) and Institut für Lebensmitteltechnologie.NRW (ILT.NRW).

References

[1] B. Vogel-Heuser, T. Bauernhansl, and M. Hompel, *Handbuch Industrie 4.0*, ser. Springer Reference Technik. Springer Berlin Heidelberg, 2017.

[2] S. K. Panda, T. Schröder, L. Wisniewski, and C. Diedrich, “Plug&Produce Integration of Components into OPC UA based data-space,” in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, Sep. 2018, pp. 1095–1100.

[3] S. K. Panda, A. Blome, L. Wisniewski, and A. Meyer, “Iot retrofitting approach for the food industry,” in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2019, pp. 1639–1642.

[4] “Industrie 4.0 Plug-and-Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation,” Federal Ministry for Economic Affairs and Energy (BMWi), Tech. Rep., June 2017.

[5] W. A. Khan, L. Wisniewski, D. Lang, and J. Jasperneite, “Analysis of the requirements for offering industrie 4.0 applications as a cloud service,” in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, June 2017, pp. 1181–1188.

[6] I. C. et. al., “Upnp device architecture,” UPnP Forum, Tech. Rep., October 2008.

[7] “OPC Unified Architecture for Devices Companion Specification,” OPC Foundation, Tech. Rep., July 2013.

[8] M. Schleipen, A. Lüder, O. Sauer, H. Flatt, and J. Jasperneite, “Requirements and concept for plug-and-work,” *at-Automatisierungstechnik*, vol. 63, no. 10, pp. 801–820, 2015.

[9] S. Profanter, K. Dorofeev, A. Zoitl, and A. Knoll, “OPC UA for plug & produce: Automatic device discovery using lds-me,” in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–8.

[10] C. Horn and J. Krüger, “A retrofitting concept for integration of machinery with legacy interfaces into cloud manufacturing architectures,” in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, Oct 2016, pp. 350–352.

[11] H. Haskamp, F. Orth, J. Wermann, and A. W. Colombo, “Implementing an opc ua interface for legacy plc-based automation systems using the azure cloud: An icps-architecture with a retrofitted rfid system,” in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, May 2018, pp. 115–121.

[12] T. Lins, R. Augusto Rabelo Oliveira, L. H. A. Correia, and J. Sa Silva, “Industry 4.0 retrofitting,” in *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*, Nov 2018, pp. 8–15.

[13] “Overview of amazon web services,” Amazon Web Services, Inc., Tech. Rep., December 2018.

[14] D. S. Linthicum, “Cloud computing changes data integration forever: What’s needed right now,” *IEEE Cloud Computing*, vol. 4, no. 3, pp. 50–53, 2017.

[15] C. Zhu, X. Li, H. Ji, and V. C. M. Leung, “Towards integration of wireless sensor networks and cloud computing,” in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, Nov 2015, pp. 491–494.

[16] “AutomationML in a Nutshell,” AutomationML consortium, Tech. Rep., November 2015.

[17] “Structure of the Administration Shell,” Federal Ministry for Economic Affairs and Energy (BMWi), Tech. Rep., April 2016.

[18] “Details of the Asset Administration Shell,” Federal Ministry for Economic Affairs and Energy (BMWi), Tech. Rep., November 2019.

[19] “Relationships between i4.0 components – composite components and smart production continuation of the development of the reference model for the i4.0 sg models and standards,” Federal Ministry for Economic Affairs and Energy (BMWi), Tech. Rep., June 2017.

[20] “Application Recommendation: AAS Representation,” AutomationML consortium, Tech. Rep., November 2019.

[21] “Opc unified architecture specification part 1: Overview and concepts,” OPC Foundation, Tech. Rep., November 2017.

[22] W. Mahnke, S. Leitner, and M. Damm, “Chapter 1 - introduction,” in *OPC Unified Architecture*. Springer, 2009, pp. 1–18.

[23] “AutomationML Whitepaper: OPC Unified Architecture Information Model for AutomationML,” AutomationML consortium, Tech. Rep., March 2016.

[24] “OPC Unified Architecture Specification Part 12: Discovery,” OPC Foundation, Tech. Rep., July 2015.